Nghiên cứu thiết kế và xây dựng phần mềm FSM-SC UAV Indoor phục vụ khảo sát hầm lò

Design and implementation of FSM-SC UAV Indoor for underground surveying

Tóm tắt

Khảo sát hầm lò bằng UAV đòi hỏi chuyển yêu cầu thu đo thành chuyển động trong hệ tọa độ cục bộ, đồng thời duy trì khả năng giao lệnh và nhận phản hồi khi phương tiện khuất tầm nhìn. Nghiên cứu xây dựng FSM-SC UAV Indoor để liên kết giao diện điều hành, dữ liệu định vị LiDAR–quán tính, quan trắc vật cản và bộ điều khiển bay. Quá trình phát triển gồm xác định yêu cầu, thiết kế kiến trúc và đối tượng dữ liệu, xây dựng luật tạo điểm đặt, lập trình giao tiếp ROS–MAVROS–PX4 và kiểm chứng theo từng mức. Nhánh Qt–goal_nav hiện thực giao mục tiêu cục bộ, giới hạn bước điểm đặt, lựa chọn hướng tránh theo ba vùng quan trắc và cập nhật phản hồi độc lập trên trạm. Sáu trường hợp kiểm tra trực tiếp thân hàm làm rõ sự tương ứng giữa biểu thức và chương trình, đồng thời xác định nhánh tiếp tục tiến khi thiếu cloud. Tập thống kê vận hành gồm 43 tệp trong 19 ngày, với 20 tệp ghi OFFBOARD và tổng thời lượng 1.609,233 s. Hai nhật ký riêng ngày 24 và 25/4/2026 cung cấp các đoạn quỹ đạo 124,54 m và 113,52 m; bảng chế độ của hai nhật ký này không ghi OFFBOARD nên được dùng để phân tích chuyển động cục bộ. Kết quả xác định sản phẩm phần mềm đã hiện thực, bằng chứng vận hành và các điều kiện cần hoàn thiện khi kiểm chứng điều hướng ở mức hệ thống.

Từ khóa: FSM-SC UAV Indoor; UAV hầm lò; điều hành; điểm đặt; LiDAR; ROS; PX4.

Abstract

Underground UAV surveys require local flight goals to be translated into vehicle motion while the operator retains command and feedback access beyond direct visual contact. This study presents the design and implementation of FSM-SC UAV Indoor, linking an operator interface, LiDAR-inertial localization, obstacle observations and a flight controller. The development process covers requirements, software architecture, data objects, setpoint generation, ROS–MAVROS–PX4 integration and layered verification. The Qt–goal_nav branch implements local goal entry, bounded setpoint steps, three-region reactive avoidance and independent vehicle feedback. Six deterministic function-level checks establish correspondence between the equations and source code and identify continued goal-directed motion when point-cloud data are unavailable. An operational summary contains 43 files over 19 days, including 20 with OFFBOARD intervals totaling 1,609.233 s. Two separate logs dated 24 and 25 April 2026 contain trajectory segments of 124.54 m and 113.52 m; their control summaries show no OFFBOARD operation, so they support local-motion analysis. The results define the implemented software, the scope of operational evidence and the remaining conditions for system-level navigation validation.

Keywords: FSM-SC UAV Indoor; underground UAV; supervisory control; setpoint; LiDAR; ROS; PX4.

1 Đặt vấn đề và mục tiêu nghiên cứu

Nhiệm vụ khảo sát hầm lò đặt ra yêu cầu đưa cảm biến đến các đoạn tuyến và vị trí có khả năng quan sát vách, nền, nóc lò để thu dữ liệu hình học và thông tin bề mặt. Khi UAV đi khuất sau chỗ ngoặt, người vận hành phải dựa vào dữ liệu truyền về để xác định vị trí phương tiện, mức thực hiện mục tiêu và tình trạng điều khiển. Việc bỏ qua một đoạn tuyến có thể làm thiếu dữ liệu tại mặt cắt cần khảo sát; việc tiếp tục phát lệnh khi quan trắc không còn phù hợp

có thể làm chuyển động khác với ý định ban đầu. Do đó, bài toán phần mềm bắt đầu từ tổ chức thực hiện nhiệm vụ thu đo và kiểm tra phản hồi trong quá trình thực hiện.

Định vị LiDAR–quán tính cung cấp tư thế và quỹ đạo cục bộ, đồng thời hỗ trợ xây dựng đám mây điểm khi thiếu GNSS. LIO-SAM tổ chức ước lượng theo đồ thị nhân tố, còn FAST-LIO2 thực hiện đăng ký trực tiếp điểm LiDAR trong khung ước lượng LiDAR–quán tính [1, 2]. Các kết quả này tạo cơ sở xác định phương tiện đang ở đâu. Ở phía điều khiển, chế độ OFFBOARD của PX4 cho phép nguồn bên ngoài gửi điểm đặt để điều khiển chuyển động; việc duy trì chế độ còn phụ thuộc dòng bản tin và các điều kiện ước lượng tương ứng [3]. Để khai thác các thành phần đó cho một tuyến khảo sát, cần xác định cách biểu diễn mục tiêu, cách lựa chọn điểm đặt và cách đối chiếu phản hồi phương tiện với thao tác điều hành.

Trong cấu hình nghiên cứu, những công việc cần phát triển gồm giao mục tiêu cùng hệ tọa độ với vị trí phản hồi; kiểm tra dữ liệu trước khi tạo bước điều khiển; sử dụng quan trắc phía trước và hai bên để lựa chọn tiến, tránh hoặc giữ; tổ chức các yêu cầu đổi chế độ và ARM/DISARM; cập nhật thông tin phương tiện trên giao diện. Đây là cơ sở xây dựng FSM-SC UAV Indoor. Giá trị của phần mềm được xem xét qua khả năng tổ chức các khâu này thành một tuyến xử lý có đầu vào, đầu ra và điều kiện thực hiện xác định.

Nghiên cứu tập trung vào nhánh trạm Qt kết hợp nút goal_nav. Phần mô tả thiết kế và thuật toán được đối chiếu với mã nguồn trạm điều khiển, nút điều hướng và cầu nối odometry. Các kết quả vận hành được phân tích theo nguồn dữ liệu và cấu hình có thể nhận diện. Mục tiêu gồm: xây dựng kiến trúc điều hành và mô hình dữ liệu; hiện thực luật tạo điểm đặt cùng các giao tiếp; xác định mức đáp ứng của từng chức năng bằng kiểm tra logic và nhật ký vận hành. Phạm vi kết luận được giới hạn theo loại bằng chứng tương ứng.

2 Yêu cầu phần mềm và cấu hình hệ thống

2.1 Từ nhiệm vụ khảo sát đến yêu cầu chức năng

Đối tượng điều hành là một UAV mang cảm biến khảo sát, sử dụng vị trí cục bộ để thực hiện mục tiêu khi thiếu GNSS. Người vận hành giao mục tiêu, quan sát phản hồi, yêu cầu thay đổi chế độ hoặc kết thúc chuyển động. Đầu vào của phần mềm gồm mục tiêu, pose cục bộ, đám mây điểm, trạng thái FCU và thao tác điều hành. Đầu ra gồm bản tin mục tiêu, điểm đặt trung gian, yêu cầu dịch vụ và thông tin hiển thị. Bảng 1 liên kết các yêu cầu với phần hiện thực và dấu hiệu cần quan sát khi kiểm chứng.

Bảng 1 Liên kết yêu cầu với kết quả phần mềm

Yêu cầu	Xử lý được xây dựng	Đầu ra và căn cứ kiểm chứng
Giao mục tiêu cục bộ	Nhập tọa độ và góc hướng; chọn tuyến gửi; tạo PoseStamped.	Bản tin mục tiêu và last_goal; đối chiếu trường có hiệu lực theo cấu hình.
Điều hướng theo mục tiêu	Giới hạn bước; xét vùng trước, trái, phải; tạo mục tiêu tránh.	Điểm đặt, trạng thái nội bộ và phản hồi vị trí theo thời gian.
Theo dõi điều kiện điều khiển	Đọc mode, pose, tuổi bản tin; kiểm tra miền XY và khóa độ cao.	Nhánh xử lý đã chọn, điểm đặt phát ra và trạng thái FCU.
Gửi yêu cầu vận hành	Gọi dịch vụ đổi chế độ, ARM/DISARM và lệnh hạ cánh.	Kết quả lời gọi và bản tin trạng thái nhận độc lập.
Hiển thị tình trạng UAV	Tập hợp dữ liệu vào StatusStore; cập nhật bản đồ và bảng trạng thái.	Giao diện, vết vị trí gần nhất và nhật ký thao tác trên trạm.

2.2 Cấu hình tích hợp và phạm vi các thành phần

Cấu hình phần mềm sử dụng ROS1, MAVROS, PX4 và nguồn định vị LiDAR–quán tính; trạm điều hành được lập trình bằng Python và PyQt5. Mã trạm hướng tới môi trường Ubuntu 20.04 và ROS Noetic. Tải LiDAR MID-360 cùng IMU cung cấp dữ liệu cho nhánh định vị và quan trắc. Hai nguyên mẫu UAV sau tích hợp tải được minh họa ở Hình 1. Vai trò trạm điều hành có thể thực hiện trên máy tính mặt đất qua mạng ROS hoặc trên máy tính gắn UAV với truy cập màn hình từ xa; vị trí cài đặt cụ thể thuộc cấu hình của từng phiên.



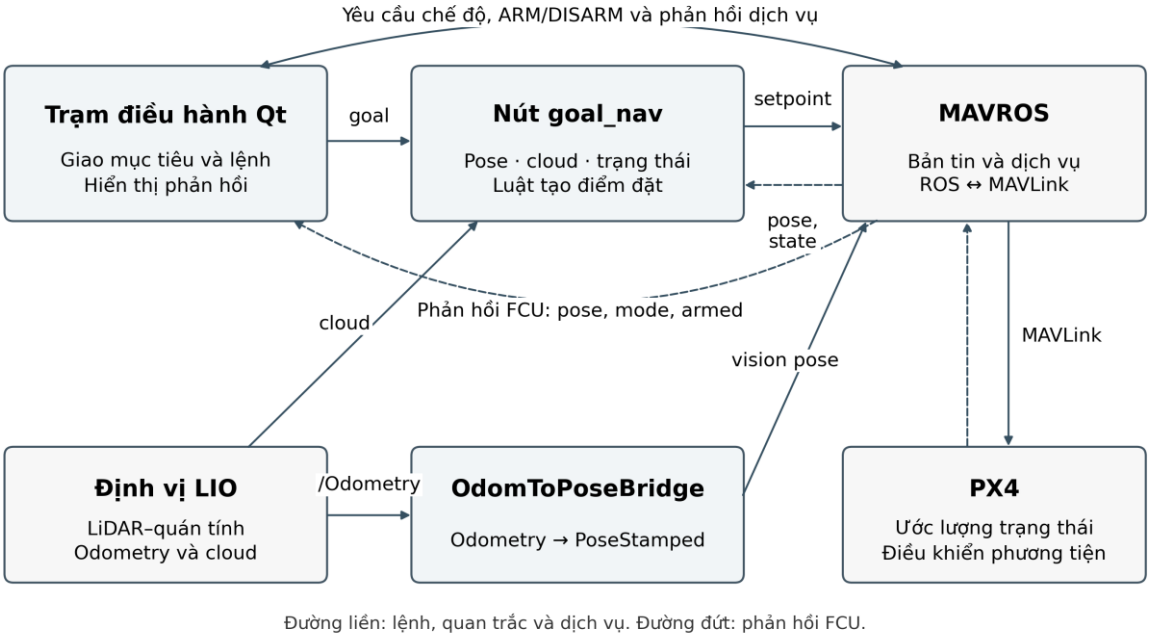
Hình 1 Hai nguyên mẫu UAV sau tích hợp tải khảo sát

Trong phạm vi nghiên cứu này, FSM-SC phát triển giao diện, tổ chức dữ liệu phản hồi, xử lý mục tiêu và luật tạo điểm đặt. Nền tảng định vị cung cấp odometry và cloud; MAVROS thực hiện cầu nối giao tiếp; PX4 thực hiện điều khiển bay theo chế độ đang hoạt động. Các tệp ULG do hệ ghi của bộ điều khiển bay tạo ra, còn PCD thuộc khâu thu và xử lý LiDAR. Việc xác định đúng thành phần tạo dữ liệu cho phép đánh giá kết quả của phần mềm điều hành trong quan hệ với toàn hệ.

3 Thiết kế kiến trúc và tổ chức dữ liệu FSM-SC

3.1 Kiến trúc chức năng và đường phản hồi

Kiến trúc được tổ chức theo đường đi của mục tiêu và đường phản hồi phương tiện (Hình 2). MainWindow tiếp nhận thao tác; RosInterface chuyển thao tác thành bản tin hoặc yêu cầu dịch vụ. Ở tuyến goal_nav, mục tiêu được xử lý cùng pose cục bộ, trạng thái FCU và cloud để lựa chọn điểm đặt. Phản hồi từ MAVROS được ghi vào StatusStore, sau đó giao diện đọc ảnh chụp trạng thái để cập nhật các thành phần hiển thị. Cách tổ chức này giữ được sự khác nhau giữa yêu cầu đã gửi và trạng thái đã nhận.



Hình 2 Kiến trúc chức năng của tuyến Qt–goal_nav đối chiếu với mã nguồn

Nguồn LiDAR–quán tính xuất /Odometry. OdomToPoseBridge sao chép tư thế sang PoseStamped và phát lên /mavros/vision_pose/pose. Nút goal_nav nhận vị trí phản hồi qua /mavros/local_position/pose; vì vậy pose từ nguồn định vị bên ngoài và pose cục bộ dùng cho điều hướng nằm trên hai kênh khác nhau. Khi tích hợp cần thống nhất hệ tọa độ, thời gian và cấu hình hợp nhất ước lượng của FCU. Tài liệu PX4 mô tả riêng yêu cầu biến đổi hệ và chuyển tiếp dữ liệu ước lượng bên ngoài [4].

Trạm còn có tuyến gửi điểm đặt trực tiếp tới MAVROS. Tuyến này phục vụ phát một điểm đặt lặp và giữ theo yêu cầu trên giao diện, trong khi tuyến goal_nav sử dụng luật phân vùng vật cản. Hai tuyến phải được nhận diện trong cấu hình đánh giá. Mã trạm có lựa chọn gửi đồng thời; do đó chỉ từ mode OFFBOARD chưa thể xác định nguồn nào đã phát điểm đặt hoặc kết luận đã có cơ chế phân quyền độc quyền giữa các nguồn.

3.2 Mô hình đối tượng và dữ liệu trong bộ nhớ

Các đối tượng được thiết kế theo trách nhiệm xử lý. StatusStore sử dụng khóa RLock để đồng bộ cập nhật từ callback ROS với thao tác đọc của giao diện. Trường state lưu phản hồi FCU; pose lưu vị trí, tư thế và hệ quy chiếu; last_goal lưu mục tiêu gần nhất. Vết hiển thị sử dụng deque giới hạn số mẫu, mặc định 600 điểm. Cấu trúc này phục vụ trạng thái hiện thời và hiển thị; độ dài bộ đệm không đại diện cho một nhật ký đầy đủ của toàn phiên.

Bảng 2 Các đối tượng chính được hiện thực trong mã nguồn

Đối tượng	Dữ liệu hoặc quan hệ chính	Vai trò
MainWindow	ros, store, signals, map_widget, timer	Nhận thao tác; đọc snapshot; cập nhật năm tab.
RosInterface	goal_pub; local_setpoint_pub; setpoint_stream_pose; subscribers	Gửi mục tiêu, điểm đặt và yêu cầu dịch vụ; nhận phản hồi.
StatusStore	data[state, pose, last_goal, …]; path; RLock	Lưu trạng thái trong bộ nhớ và vết vị trí gần nhất.

Đối tượng	Dữ liệu hoặc quan hệ chính	Vai trò
GuiSignals	log(str); command_done(str, bool)	Chuyển thông báo từ luồng xử lý về giao diện.
MapWidget	path; current; goal	Vẽ vết vị trí và mục tiêu gần nhất trong mặt phẳng cục bộ.
RvizGoalToPX4Avoid	current_pose; goal_pose; avoid_target; mode; hold_xy; hold_altitude	Tính điểm đặt và tổ chức các nhánh điều hướng.

Thông tin trạng thái được phân biệt ở ba lớp. Mode OFFBOARD/POSCTL thuộc FCU; AUTO/AVOID/HOLD thuộc nút điều hướng; RUNNING/IDLE và bộ đếm waypoint trong phiên ghi hình thuộc giao diện web. Mã trạm Qt được khảo sát quản lý mục tiêu gần nhất, chưa có bộ thực thi danh sách waypoint tương ứng với giao diện web. Vì vậy trạng thái của các lớp được phân tích theo đúng ý nghĩa và nguồn tạo, tránh dùng chúng thay thế cho nhau.

3.3 Thiết kế các giao tiếp và chuỗi thực hiện lệnh

Chuỗi giao mục tiêu bắt đầu khi cmd_send_goal đọc X, Y, Z và góc hướng, chuyển góc từ độ sang radian và gọi publish_goal. Bản tin PoseStamped đi qua /move_base_simple/goal tới goal_cb của nút điều hướng. timer_cb dùng dữ liệu mới nhất để chọn điểm đặt, sau đó phát lên /mavros/setpoint_position/local. Song song, _state_cb và _pose_cb của trạm cập nhật StatusStore; refresh đọc snapshot để hiển thị. Tần số cập nhật giao diện trong cấu hình là 2 Hz; tần số điểm đặt của nút điều hướng là 20 Hz. Đây là tần số cấu hình, chưa thay thế phép đo độ trễ hoặc tần số thực nhận.

Bảng 3 Các kênh giao tiếp của tuyến điều hành

Giao tiếp	Kiểu dữ liệu	Nguồn và nơi nhận
/move_base_simple/goal	PoseStamped	RosInterface → goal_cb
/mavros/local_position/pose	PoseStamped	MAVROS → goal_nav và trạm
/mavros/state	State	MAVROS → goal_nav và trạm
/cloud_registered	PointCloud2	Khâu đăng ký cloud → goal_nav theo launch
/mavros/setpoint_position/local	PoseStamped	goal_nav hoặc tuyến trực tiếp → MAVROS
/mavros/vision_pose/pose	PoseStamped	OdomToPoseBridge → MAVROS
/mavros/set_mode; /mavros/cmd/arming	SetMode; CommandBool	RosInterface → dịch vụ MAVROS

Các lời gọi dịch vụ được xử lý trên luồng riêng để giao diện tiếp tục nhận dữ liệu. Kết quả mode_sent hoặc success được ghi vào nhật ký giao diện; mode và armed vẫn được cập nhật từ bản tin FCU. Hàm hạ cánh của nhánh Qt gọi set_mode("AUTO.LAND"). Nút dừng stream chỉ tắt luồng điểm đặt trực tiếp do trạm phát; thao tác đó không tự đổi mode FCU và không dừng nút goal_nav. Những khác biệt này được đưa vào mô tả giao tiếp để xác định đúng kết quả của từng thao tác.

4 Xây dựng thuật toán và lập trình tích hợp

4.1 Biểu diễn mục tiêu và tạo điểm đặt giới hạn bước

Ký hiệu x_k là vị trí ngang phản hồi tại chu kỳ k, g là mục tiêu cuối và t_k là mục tiêu đang được hàm tạo bước sử dụng. Khi đi trực tiếp, $t_k = g$; khi tránh, t_k là avoid_target. Các tọa độ được biểu diễn bằng mét trong cùng hệ. Hàm clamp_step_target thực hiện:

$$d_k = \|t_k - x_k\|_2, \quad s_k = min(\text{step_xy}, \text{max_goal_step}, d_k) \quad (1)$$

$$x_{sp,k} = x_k + s_k \frac{t_k - x_k}{d_k} \quad (2)$$

Khi khoảng cách tới mục tiêu truyền vào hàm nhỏ hơn 10^{-6} m, hàm trả vị trí ngang hiện tại để tránh phép chia gần zero. Trong vòng xử lý chính, điều kiện khoảng cách ngang tới g nhỏ hơn goal_tolerance được xét trước khi tạo bước; nút xóa mục tiêu và cập nhật điểm giữ theo pose hiện tại. Ngưỡng goal_tolerance = 0,25 m là điều kiện chuyển trạng thái, còn sai lệch đạt mục tiêu của phương tiện phải tính riêng từ dữ liệu đồng bộ.

Mã 1 trích phần tính bước trong hàm clamp_step_target. Sự tương ứng giữa biểu thức và thân hàm được kiểm tra bằng các trường hợp đầu vào xác định ở mục 5.1. Việc giới hạn bước điểm đặt không trực tiếp đặt vận tốc bay bằng tích bước và tần số phát, vì mỗi bước được tính lại từ vị trí phản hồi và được FCU xử lý tiếp.

Mã 1 Phần tính bước điểm đặt trong clamp_step_target

```
dx, dy = tx - cx, ty - cy
dist = math.hypot(dx, dy)
if dist < 1e-6:
   return cx, cy
step = min(self.step_xy, self.max_goal_step, dist)
return cx + step * dx / dist, cy + step * dy / dist
```

4.2 Phân vùng quan trắc và lựa chọn hướng tránh

Hướng xét cloud được xác định theo mục tiêu đang xử lý. Đặt e_k là vector đơn vị từ x_k tới t_k và n_k là hướng ngang trái vuông góc với e_k. Với điểm cloud q_i đã cùng hệ tọa độ với pose, các thành phần theo hướng tiến và hướng ngang được xác định bởi:

$$e_k = \frac{t_k - x_k}{d_k}, \quad n_k = (-e_{y,k}, e_{x,k}) \quad (3)$$

Khi khoảng cách ngang tới mục tiêu đang xét nhỏ hơn 10^{-6} m, get_goal_direction sử dụng hướng dự phòng $e_k = (1; 0)$ và hướng vuông góc $n_k = (0; 1)$.

$$a_i = (q_{i,xy} - x_k) \cdot e_k, \quad b_i = (q_{i,xy} - x_k) \cdot n_k \quad (4)$$

Hàm analyze_cloud loại điểm nằm trong bán kính bỏ qua quanh UAV, lọc theo chênh cao được cấu hình và chỉ xét điểm có thành phần dọc từ 0 đến look_ahead_max. Vùng trước có $|b_i|$ < front_half_width; vùng trái và phải nằm giữa front_half_width và side_half_width với dấu của b_i tương ứng. Khoảng cách phía trước là giá trị nhỏ nhất của a_i; khoảng cách hai bên là giá trị nhỏ nhất của căn bậc hai $a_i^2 + b_i^2$ trong từng vùng. Đây là các đại lượng dùng trong luật chọn hướng, chưa phải khoảng cách nhỏ nhất từ toàn bộ thân UAV tới bề mặt.

Ở trạng thái AUTO, khi phía trước đủ thoáng, nút phát bước tới g. Nếu phía trước bị chặn, choose_avoid_direction so sánh khoảng cách hai bên với side_clearance. Chỉ một bên vượt ngưỡng thì chọn bên đó; cả hai vượt ngưỡng thì chọn bên có khoảng cách lớn hơn, trường hợp

bằng nhau chọn trái; cả hai không vượt ngưỡng thì phát điểm giữ. Mục tiêu tránh được tạo theo:

$$t_{avoid} = x_k + l_f e_k \pm l_s n_k \quad (5)$$

Trong cấu hình được khảo sát, l_f = avoid_forward_step = 0,10 m và l_s = side_step = 0,80 m; dấu cộng ứng với tránh trái. Mục tiêu này tiếp tục đi qua phép giới hạn bước (1)–(2). Ở chu kỳ vừa chọn hướng tránh, chương trình phát điểm giữ rồi kết thúc chu kỳ; các chu kỳ tiếp theo mới tạo bước tới avoid_target. Khi đang tránh, hướng xét quan trắc được cập nhật theo mục tiêu tránh, đồng thời vật cản phía trước được kiểm tra lại.

4.3 Trạng thái xử lý và tham số thực thi

Nút điều hướng hiện thực các trạng thái IDLE, PRE_OFFBOARD_HOLD, AUTO, AVOID_LEFT, AVOID_RIGHT, HOLD và EMERGENCY_HOLD. Trước các nhánh tiến hoặc tránh, timer_cb kiểm tra lần lượt sự có mặt của pose, tuổi bản tin nhận, miền vị trí XY và mode FCU. Khi vào OFFBOARD, chương trình khóa điểm và độ cao giữ. Nếu không cho phép thay đổi độ cao nhưng vị trí phản hồi tăng vượt ngưỡng cấu hình, nút xóa mục tiêu và phát điểm đặt theo pose hiện tại trong nhánh EMERGENCY_HOLD.

Khi đã đạt đích, điểm giữ được cập nhật theo vị trí đạt đích. Ở các nhánh khác, hold_sp có thể là điểm đã khóa lúc vào OFFBOARD; do đó hành vi phát điểm giữ cần được phân biệt với phát chính pose hiện tại. Nếu phía trước rất gần, nhánh tránh xét lại hướng và có thể đặt lại bộ đếm. Khi cloud hợp lệ, điều kiện kết thúc tránh theo nhánh thông thường là khoảng cách phía trước vượt safe_front và bộ đếm tránh đã hết. HOLD do vật cản khi mục tiêu còn tồn tại được xét lại ở chu kỳ sau, không biểu thị hủy nhiệm vụ vĩnh viễn.

Bảng 4 Tham số mặc định và tham số ghi đè trong cấu hình

Nhóm tham số	Mặc định trong mã	Tệp launch được khảo sát
Tần số phát; ngưỡng đạt đích	20 Hz; 0,25 m	20 Hz; 0,25 m
step_xy; max_goal_step	0,15 m; 0,20 m	0,20 m; 0,30 m
Miền X và Y; pose_timeout	−10 đến 10 m; 0,5 s	−100 đến 100 m; 0,5 s
Cloud; cloud_timeout	/livox/lidar; 0,3 s	/cloud_registered; 0,7 s
z_window_half	0,45 m	999 m
Bán kính bỏ qua; tầm xét	0,30 m; 2,00 m	0,30 m; 2,00 m
Nửa rộng vùng trước; vùng bên	0,35 m; 1,20 m	0,35 m; 1,20 m
safe_front; hard_stop_front	1,20 m; 0,70 m	1,20 m; 0,70 m
side_clearance; side_step	0,90 m; 0,80 m	0,90 m; 0,80 m
Bước tiến khi tránh; bộ đếm	0,10 m; 20 chu kỳ	0,10 m; 20 chu kỳ

Bảng 4 mô tả hai nguồn tham số trong mã và tệp khởi chạy, chưa xác định cấu hình của mọi nhật ký vận hành. Giá trị z_window_half = 999 m làm vùng lọc độ cao rất rộng; cấu hình này cần được phân biệt với mặc định ±0,45 m. Tham số allow_altitude_change mặc định bằng False nên độ cao giữ được xác định theo logic khóa, dù target_altitude có giá trị khai báo 1,5 m. Góc hướng nhập trên giao diện được mã hóa trong bản tin mục tiêu, nhưng make_pose của nút điều hướng tạo quaternion đơn vị; vì vậy nhánh này chưa hiện thực bám đầy đủ mục tiêu XYZ và góc hướng như các trường nhập gợi ý.

4.4 Kết quả lập trình và điều kiện tích hợp

Kết quả lập trình gồm trạm Qt năm tab, bộ lưu trạng thái trong bộ nhớ, lớp giao tiếp ROS, nút điều hướng và cầu nối dữ liệu định vị. Các tab tổ chức thông tin điều khiển, vị trí và quan trắc, dữ liệu môi trường, cấu hình kết nối và nhật ký theo chức năng trong mã. Mô đun môi trường có các trường khí và vi khí hậu để nhận dữ liệu khi có nguồn cung cấp; kết quả định lượng của cảm biến môi trường nằm ngoài phần kiểm chứng điều hướng trong nghiên cứu này.

Ba điều kiện tích hợp có ảnh hưởng trực tiếp đến ý nghĩa lệnh. Thứ nhất, cloud và pose phải cùng hệ trước khi tính hiệu tọa độ ở (4), vì analyze_cloud không tự thực hiện biến đổi TF. Thứ hai, tuổi dữ liệu cần được xét theo nguồn đo: OdomToPoseBridge phát lại pose cuối với dấu thời gian hiện tại, nên việc còn nhận bản tin chưa đủ xác nhận nguồn LiDAR–quán tính còn cập nhật. Thứ ba, nguồn phát điểm đặt phải được xác định theo cấu hình, vì trạm và goal_nav có thể cùng xuất vào một topic. Các điều kiện này là cơ sở giới hạn phạm vi đánh giá của bản dựng và lựa chọn phép kiểm tra ở phần tiếp theo.

5 Kiểm chứng phần mềm và phân tích kết quả vận hành

5.1 Phương pháp kiểm chứng và kiểm tra logic thuật toán

Kiểm chứng được thực hiện ở hai mức. Mức thứ nhất kiểm tra sự tương ứng giữa thuật toán và chương trình bằng cách tách trực tiếp các thân hàm từ cây cú pháp của mã nguồn, sau đó thực thi với đầu vào xác định. Các đối tượng pose, cloud, thời gian và bộ thu điểm đặt được cung cấp trong bộ nhớ; không khởi tạo ROS, không kết nối FCU và không sử dụng mô hình động lực học bay. Mức thứ hai phân tích kết quả vận hành đã ghi, gồm thống kê 43 nhật ký, bộ phân tích hai nhật ký tháng 4/2026, một phiên ghi hình giao diện web và ba phiên thu PCD. Tập 43 nhật ký và ba phiên PCD được sử dụng ở mức chỉ tiêu tổng hợp. Các nguồn được trình bày riêng vì chưa có định danh phiên chung để quy toàn bộ kết quả cho cùng một cấu hình.

Bảng 5 Kết quả sáu trường hợp kiểm tra trực tiếp thân hàm

Ca	Đầu vào kiểm tra	Đầu ra quan sát và ý nghĩa
K01	x = (0; 0), t = (3; 4); step_xy = 0,40 m; max_goal_step = 0,20 m.	Điểm đặt (0,12; 0,16), bước dài 0,20 m. Xác nhận giới hạn max_goal_step với tham số thử.
K02	Mục tiêu cách 0,10 m; ngưỡng đạt đích 0,25 m.	timer_cb chuyển HOLD, xóa goal và phát pose hiện tại. Kiểm tra đạt đích được ưu tiên.
K03	d_front = 0,60 m; d_left = 1,131 m; d_right = 0,707 m.	Chọn AVOID_LEFT; mục tiêu tránh (0,10; 0,80); bộ đếm 20. Chu kỳ đầu phát điểm giữ.
K04	Phía trước bị chặn; d_left = d_right = 0,707 m; ngưỡng bên 0,90 m.	Chọn HOLD và phát điểm giữ. Không bên nào vượt ngưỡng khoảng trống.
K05	Không có cloud; mục tiêu (3; 0); bước mặc định 0,15 m.	Phát điểm đặt (0,15; 0; 1,5). Xác nhận bản dựng vẫn tiến khi thiếu cloud.
K06	Đang AVOID_LEFT; cloud tuổi 0,31 s, vượt mặc định 0,30 s.	Chuyển AUTO và phát (0,15; 0; 1,5). Xác nhận nhánh cloud quá hạn khi đang tránh.

Các ca K02–K06 dùng pose và điểm giữ trùng nhau tại (0; 0; 1,5) m, mode FCU OFFBOARD và các tham số mặc định, trừ điều kiện được nêu riêng. Cloud của K03 gồm các điểm (0,60; 0; 1,50), (0,80; 0,80; 1,50) và (0,50; −0,50; 1,50) m; K04 thay điểm trái bằng

(0,50; 0,50; 1,50) m. Mục tiêu ngang của hai ca là (3; 0) m. K01 thay đổi hai giới hạn bước để làm xuất hiện nhánh max_goal_step. Bốn ca đầu xác nhận phép giới hạn bước, thứ tự kiểm tra đạt đích và lựa chọn tránh/giữ. Hai ca cuối xác định đúng một giới hạn của phiên bản; chúng không được tính là các ca đáp ứng yêu cầu xử lý mất quan trắc.

Kết quả mức hàm chứng minh các quy tắc đã được hiện thực và xác định đầu ra trong các điều kiện thử. Chuyển động thực của UAV, khoảng cách thân phương tiện tới vật cản và chất lượng bám tuyến còn phụ thuộc chuỗi tích hợp và điều kiện bay, vì vậy được xem xét bằng dữ liệu vận hành riêng.

5.2 Thống kê chế độ trong tập nhật ký vận hành

Bảng 6 Chế độ điều khiển trong tập thống kê 43 nhật ký

Chỉ tiêu	Kết quả
Quy mô tập thống kê	43 tệp trong 19 ngày
Tệp ghi nhận OFFBOARD	20 tệp
Tệp có thời lượng OFFBOARD ít nhất 10 s	15 tệp
Tệp có chuỗi AUTO_TAKEOFF–OFFBOARD–AUTO_LAND	4 tệp
Tổng thời lượng OFFBOARD cộng dồn	1.609,233 s, tương đương 26 phút 49,233 giây

Trong tập thống kê, 20/43 tệp có OFFBOARD, tương ứng 46,51% số tệp; trong nhóm này, 15/20 tệp có thời lượng OFFBOARD ít nhất 10 s. Bốn tệp ghi nhận chuỗi chuyển chế độ qua cất cánh tự động, OFFBOARD và hạ cánh tự động. Kết quả cho thấy hệ thống đã có các khoảng vận hành nhận điểm đặt bên ngoài. Tổng thời gian 1.609,233 s được cộng từ nhiều khoảng ghi nhận, không biểu diễn một chuyến bay liên tục.

Đơn vị thống kê ở Bảng 6 là tệp, trong khi đơn vị đánh giá hoàn thành phải là nhiệm vụ có mục tiêu và tiêu chí xác định. Do đó tỷ lệ tệp có OFFBOARD chưa thể chuyển thành tỷ lệ nhiệm vụ thành công. Việc quy kết kết quả cho goal_nav đòi hỏi cấu hình phát lệnh của phiên, các mục tiêu đã giao và chuỗi điểm đặt tương ứng. Thống kê mode cung cấp bằng chứng về chế độ vận hành; nó không tự xác định nguồn điểm đặt trong những cấu hình có nhiều tuyến điều khiển.

5.3 Phân tích quỹ đạo cục bộ theo các đoạn liên tục

Hai mẫu quỹ đạo được nhận diện bằng tên log_1_2026-4-24-18-26-02.ulg và log_2_2026-4-25-16-09-36.ulg trong bộ phân tích quỹ đạo. Các bảng ulg_metrics_summary.csv và ulg_segmented_metrics_filtered.csv cho phép đối chiếu chế độ, số mẫu, thời gian và các chỉ tiêu theo đoạn. Bảng tổng hợp của cả hai mẫu ghi flag_control_manual_enabled_pct = 100% và flag_control_offboard_enabled_pct = 0%. Vì vậy, chúng được dùng để phân tích chuyển động ước lượng và gián đoạn tọa độ; các mẫu này không được dùng làm bằng chứng thực hiện mục tiêu bằng goal_nav trong OFFBOARD.

Đối với đoạn liên tục gồm n vị trí p_i theo thời gian t_i, chiều dài tích lũy L, độ dịch chuyển đầu–cuối D và tốc độ dọc quỹ đạo trung bình được định nghĩa:

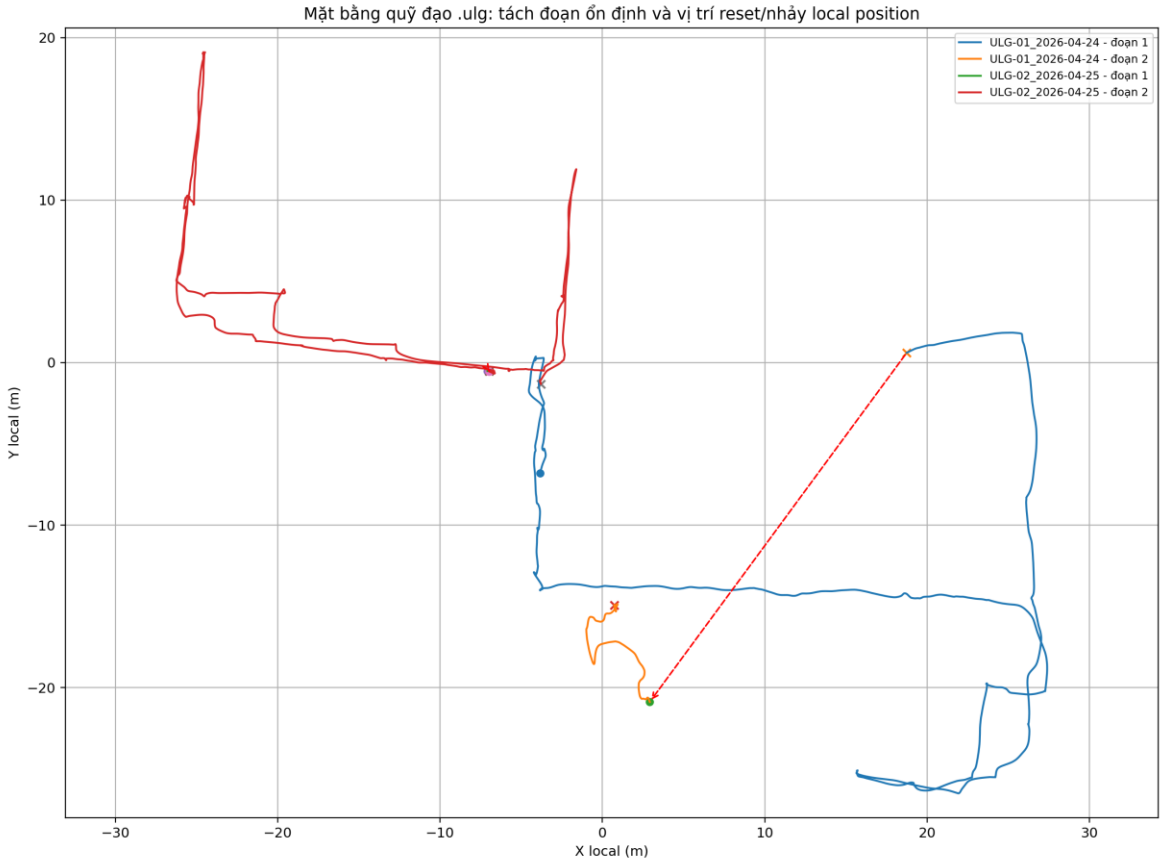
$$L = \sum_{i=2}^{n} \|p_i - p_{i-1}\|_2, \quad D = \|p_n - p_1\|_2 \qquad (6)$$

$$T = t_n - t_1, \quad v_{path} = \frac{L}{T}, \quad \rho = 100\frac{D}{L}\,\% \qquad (7)$$

Hai mẫu được đối chiếu qua các bảng CSV và hình quỹ đạo đã xuất. Các giá trị L và D ở Bảng 7 lấy từ bảng phân đoạn nguồn; T, L/T và tỷ lệ D/L được kiểm tra lại từ số liệu tổng hợp. Nghiên cứu chưa tái lập việc tách đoạn từ chuỗi pose gốc. Hình 3 sử dụng hình quỹ đạo đã phân tích của cùng nguồn; thời gian là thời gian tương đối của từng log. Việc tách các bước nhảy tọa độ giúp tránh cộng gián đoạn ước lượng vào chiều dài chuyển động.

Bảng 7 Chỉ tiêu quỹ đạo theo đoạn liên tục của hai log tháng 4 năm 2026

Đoạn	Số pose	T (s)	L (m)	D (m)	L/T (m/s)
ULG01–1	2.721	271,896	124,544	23,785	0,458
ULG01–2	294	29,311	14,345	6,452	0,489
ULG02–1	93	9,107	0,545	0,495	0,060
ULG02–2	3.064	306,300	113,520	3,262	0,371



Hình 3 Các đoạn quỹ đạo XY: ULG01–1 màu xanh lam; ULG01–2 màu cam; ULG02–1 màu xanh lục; ULG02–2 màu đỏ. Nét đứt đánh dấu bước nối bị loại; hai log có hệ cục bộ riêng, không biểu diễn chung một bản đồ.

ULG01–1 ghi 2.721 pose trong 271,896 s, chiều dài 124,544 m và độ dịch chuyển đầu–cuối 23,785 m. ULG02–2 ghi 3.064 pose trong 306,300 s, chiều dài 113,520 m và độ dịch chuyển đầu–cuối 3,262 m. Tỷ lệ D/L lần lượt là 19,10% và 2,87%. Đoạn thứ hai có đường đi tích lũy lớn nhưng kết thúc gần điểm đầu; hình học này chưa xác nhận chức năng tự trở về khi không có sự kiện kích hoạt và điều kiện kết thúc tương ứng.

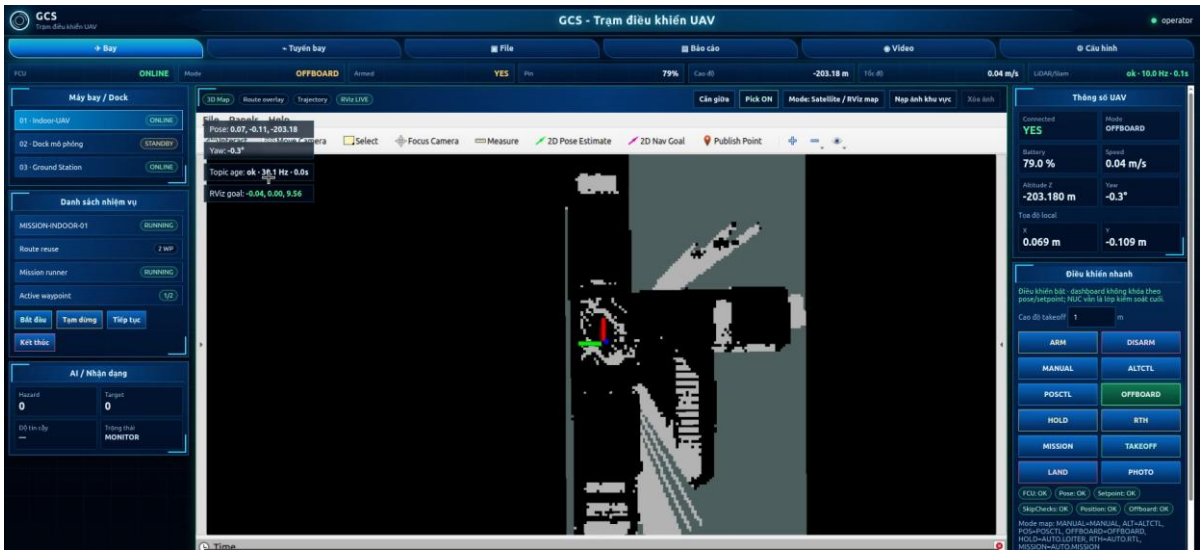
Hai đoạn ngắn có ý nghĩa khác nhau: ULG01–2 dài 14,345 m trong 29,311 s, còn ULG02–1 dài 0,545 m trong 9,107 s. Tốc độ L/T của chúng lần lượt là 0,489 m/s và 0,060

m/s. Đoạn dịch chuyển nhỏ chưa đủ xác nhận chế độ giữ vị trí, vì còn cần điểm giữ và độ phân tán quanh điểm đó. Tốc độ ở đây được định nghĩa theo L/T; không thay bằng trường speed_mean của bảng nguồn khi chưa xác định cùng cách tổng hợp.

Ảnh hưởng của gián đoạn thể hiện rõ khi đối chiếu chiều dài toàn chuỗi với tổng các đoạn. ULG01 có chiều dài toàn chuỗi 165,574 m, còn hai đoạn cộng 138,889 m; phần chênh 26,685 m trùng với bước nhảy lớn nhất được bảng nguồn ghi nhận. ULG02 có các giá trị tương ứng 340,222 m, 114,066 m và 226,156 m. Vì vậy, phép cộng khoảng cách giữa mọi cặp mẫu liên tiếp có thể tạo kết quả sai về quy mô đường đi nếu không xét tính liên tục của hệ tọa độ. Đây là căn cứ dữ liệu cho yêu cầu kiểm tra vị trí trước khi diễn giải quỹ đạo và phản hồi điều khiển.

5.4 Quan sát giao diện và sản phẩm thu cục bộ

Phiên ghi hình sử dụng giao diện web sáu tab, khác với hiện thực Qt đã phân tích mã nguồn. Ở phút 03:10, trạm đã khai báo hai điểm tuyến nhưng tiến trình còn IDLE. Đến phút 06:30, giao diện hiển thị OFFBOARD, Armed YES và RUNNING tại điểm 1/2 (Hình 4). Ảnh này minh họa cách tổ chức thông tin điều hành trong phiên web; nó không được dùng để xác nhận giao diện Qt hoặc xác định thuật toán phát điểm đặt của phiên.



Hình 4 Giao diện web tại phút 06:30 với phản hồi FCU và tiến trình điểm tuyến. Đây là cấu hình giao diện của phiên ghi hình, tách biệt với nhánh Qt được phân tích mã nguồn.

Tại phút 07:40, trạm còn hiển thị RUNNING ở điểm 2/2 trong khi FCU báo POSCTL và Armed NO. Tình huống này xác định giới hạn của việc dùng tiến trình phía trạm để kết luận trạng thái thực thi của phương tiện. Kết quả được sử dụng để đánh giá yêu cầu cập nhật trạng thái khi mode hoặc armed thay đổi; nó chưa chứng minh nhiệm vụ đã hoàn thành hoặc cơ chế đồng bộ đã được kiểm chứng sau sửa.

Bảng 8 Quỹ đạo và sản phẩm PCD ghi trong ba phiên thu cục bộ

Bắt đầu phiên	Armed ghi nhận	Đường cục bộ (m)	Số điểm PCD	Trạng thái khi lưu
14:39:47	273 s	66,8	841.768	RTH chờ chuyển OFFBOARD
15:04:14	315 s	66,3	1.062.796	RTH tạm dừng do người lái điều khiển

Bắt đầu phiên	Armed ghi nhận	Đường cục bộ (m)	Số điểm PCD	Trạng thái khi lưu
15:27:07	121 s đến lần tự lưu	22,7	733.845	IDLE; vết gồm 32 điểm

Ba phiên lưu tổng cộng 2.638.409 điểm LiDAR. Hai phiên có chiều dài gần nhau, 66,8 m và 66,3 m, nhưng trạng thái tại lúc lưu khác nhau: một phiên chờ chuyển OFFBOARD để trở về, phiên kia tạm dừng do người lái điều khiển. Kết quả cho thấy cần gắn sản phẩm thu với diễn biến điều hành của từng phiên. Số điểm phản ánh lượng bản ghi đã lưu, trong khi độ phủ và độ chính xác bản đồ cần các phép đánh giá riêng. Các trạng thái trong Bảng 8 chưa cung cấp căn cứ tính tỷ lệ RTH hoàn thành.

5.5 Thảo luận theo yêu cầu và mức bằng chứng

Đối với yêu cầu giao mục tiêu và tạo điểm đặt, mã nguồn xác định được đường xử lý từ trường nhập tới bản tin, luật giới hạn bước và lựa chọn hướng tránh. Kiểm tra mức hàm xác nhận một số nhánh điển hình và chỉ ra các trường hợp không đáp ứng xử lý mất quan trắc. Đây là kết quả về hiện thực thuật toán. Đối với yêu cầu phản hồi, mô hình StatusStore, callback và kênh trạng thái FCU xác định được cách tiếp nhận và hiển thị. Các quan sát vận hành cho thấy cần tiếp tục ràng buộc ý nghĩa tiến trình nhiệm vụ với trạng thái thực nhận.

Bốn giới hạn ảnh hưởng trực tiếp đến kiểm chứng ở mức hệ thống. Một là, cloud không có hoặc quá hạn vẫn dẫn tới bước đi trực tiếp; cloud đọc được nhưng không có điểm thuộc vùng xét cũng giữ các khoảng cách ở vô cực. Hai là, dấu thời gian phát lại từ cầu nối và tuổi bản tin nhận chưa biểu diễn đầy đủ tuổi quan trắc gốc. Ba là, phép trừ tọa độ cloud–pose và các tuyến phát điểm đặt cần được ràng buộc bởi cấu hình tích hợp. Bốn là, các tập nhật ký, quỹ đạo tháng 4 và phiên giao diện web chưa có liên kết chung đủ để quy kết mọi chuyển động cho nhánh Qt–goal_nav.

Các giới hạn này xác định phạm vi của kết quả: phần mềm đã có các thành phần giao mục tiêu, xử lý điều hướng phản ứng và theo dõi phản hồi; tập vận hành có dữ liệu chế độ, quỹ đạo và PCD. Đánh giá hoàn thành tuyến bằng đúng bản dựng cần ghép mục tiêu, điểm đặt, pose, cloud và trạng thái theo cùng phiên. Khi kiểm chứng chức năng tránh vật cản, chỉ tiêu phải gắn với phạm vi quan sát và kích thước phương tiện; khi đánh giá bám tuyến, cần phân biệt sai lệch theo pose ước lượng với sai số so với tham chiếu độc lập.

6 Kết luận

Nghiên cứu đã xác định nhu cầu xây dựng FSM-SC UAV Indoor từ nhiệm vụ điều hành UAV mang cảm biến khảo sát trong môi trường thiếu GNSS và khuất tầm nhìn. Giải pháp được cụ thể hóa bằng kiến trúc giao mục tiêu–tạo điểm đặt–nhận phản hồi, mô hình đối tượng trong bộ nhớ và các giao tiếp ROS–MAVROS–PX4. Kết quả lập trình gồm trạm Qt, bộ lưu trạng thái, nút điều hướng theo mục tiêu và cầu nối dữ liệu định vị. Luật điều hướng hiện thực giới hạn bước, phân vùng quan trắc, chọn hướng tránh và các nhánh kiểm tra pose, mode cùng độ cao.

Sáu trường hợp kiểm tra thân hàm xác nhận sự tương ứng của các biểu thức với một số nhánh chương trình và làm rõ hành vi khi thiếu dữ liệu vật cản. Ở mức vận hành, tập 43 tệp có 20 tệp ghi OFFBOARD, tổng thời lượng 1.609,233 s; ba phiên thu lưu 2.638.409 điểm LiDAR. Hai nhật ký tháng 4/2026 cung cấp các đoạn quỹ đạo 124,544 m và 113,520 m, đồng thời cho thấy ảnh hưởng lớn của bước nhảy tọa độ khi tính chiều dài. Do bảng chế độ hai mẫu

này không ghi OFFBOARD, chúng được sử dụng cho phân tích chuyển động cục bộ, không được quy thành kết quả điều hướng của goal_nav.

Đóng góp của nghiên cứu là thiết kế và hiện thực tuyến điều hành có thể truy nguyên từ yêu cầu tới đối tượng, thuật toán và giao tiếp, đồng thời xác định mức bằng chứng của từng kết quả. Việc hoàn thiện tiếp tập trung vào xử lý quan trắc không hợp lệ, bảo toàn tuổi dữ liệu nguồn, thống nhất hệ tọa độ và nguồn phát điểm đặt, cùng cập nhật tiến trình theo phản hồi FCU. Các chức năng bám tuyến, tránh vật cản và tự trở về cần được đánh giá trên những phiên có đủ lệnh và phản hồi đồng bộ trước khi công bố độ chính xác hoặc tỷ lệ hoàn thành.

Tài liệu tham khảo

[1] Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C. và Rus, D. (2020). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. IEEE/RSJ International Conference on Intelligent Robots and Systems, 5135–5142. DOI: 10.1109/IROS45743.2020.9341176.

[2] Xu, W., Cai, Y., He, D., Lin, J. và Zhang, F. (2022). FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. IEEE Transactions on Robotics, 38(4), 2053–2073. DOI: 10.1109/TRO.2022.3141876. https://arxiv.org/abs/2107.06829

[3] PX4 Development Team. Offboard Mode. PX4 User Guide, nhánh main. https://docs.px4.io/main/en/flight_modes/offboard. Truy cập ngày 19/11/2026.

[4] PX4 Development Team. Using Vision or Motion Capture Systems for Position Estimation. PX4 User Guide, nhánh main. https://docs.px4.io/main/en/ros/external_position_estimation. Truy cập ngày 19/11/2026.