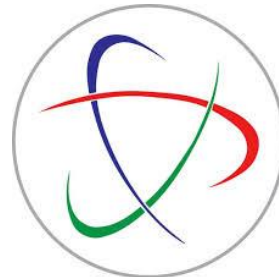


ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO CUỐI KÌ
HỌC PHẦN: LẬP TRÌNH MẠNG

ĐỀ TÀI:
MÔ PHỎNG VÀ ĐIỀU KHIỂN USV
TRÊN NỀN TẢNG PX4 VÀ MAVLINK

GVHD: TS. NGUYỄN VĂN HIẾU

SVTH: LÊ THANH TRÀ - 106220273
TRẦN ĐẠI TÍN - 106220272
NGUYỄN XUÂN TRƯỜNG - 106220275

Lớp: 22KTMT2

Đà Nẵng, 11/2025

MỤC LỤC

1. Phân công công việc	1
2. Các nội dung đã thực hiện	1
2.1. Lý thuyết tổng quan	1
2.2. Mô hình hệ thống / mô hình mạng của dự án	2
2.3. Mô tả kịch bản mô phỏng	9
2.4. Điều khiển waypoint tự động (Mission Control) - Điểm mới của nhóm	11
3.1. Các metrics sử dụng để đánh giá hệ thống	11
3.2. Mô tả các file log kết quả.	12
3.3. Kết quả chạy thực nghiệm và so sánh	15
3.3.1. Kết quả đo khi thuyền ở vị trí home khởi chạy từ PX4	15
3.3.2. Kết quả khi thuyền di chuyển theo mission	18
4. Kết luận	22

1. Phân công công việc

HỌ VÀ TÊN	NHIỆM VỤ CHÍNH	CHI TIẾT	TỶ LỆ
LÊ THANH TRÀ	Backend + Server MAVLink/UDP	<ul style="list-style-type: none">- Thiết kế và lập trình server Python (server.py) nhận dữ liệu từ PX4 qua MAVLink.- Xử lý dữ liệu telemetry, mission, GPS, battery.- Ghi log (latency, packet loss, PX4 data).- Hướng dẫn kết nối UDP với PX4.	34%
TRẦN ĐẠI TÍN	PX4 Simulation + MAVLink	<ul style="list-style-type: none">- Thiết lập và cấu hình PX4 trên mô phỏng Gazebo.- Thử nghiệm kết nối MAVLink, gửi/nhận dữ liệu từ USV.- Kiểm tra mission upload.	33%
NGUYỄN XUÂN TRƯỜNG	Testing + Dashboard + Logging	<ul style="list-style-type: none">- Tạo dashboard hiển thị telemetry (Flask + HTML/JS).- Kiểm tra logging, đo latency và packet loss.- Thử nghiệm API endpoints, validate dữ liệu.	33%

2. Các nội dung đã thực hiện

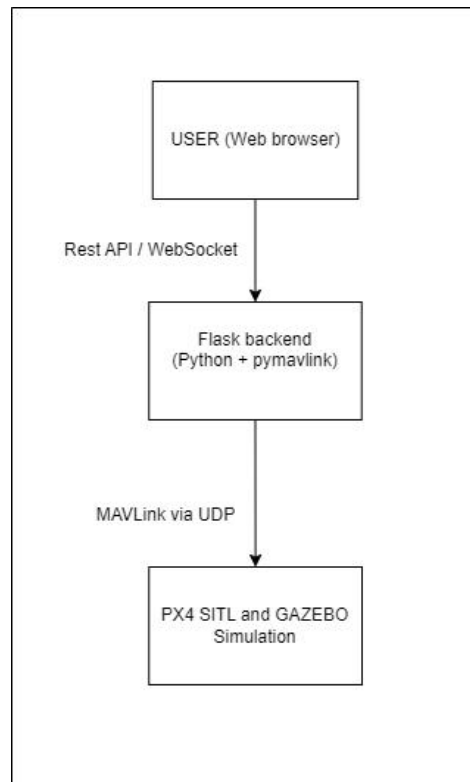
2.1. Lý thuyết tổng quan

- **PX4** là một phần mềm điều khiển mã nguồn mở dùng cho các phương tiện tự hành (UAV, USV), cung cấp bộ công cụ điều khiển và module điều hướng, tiêu chuẩn phần mềm và phần cứng cũng như các giao thức giao tiếp, mô phỏng SITL.
- **MAVLink** (Micro Air Vehicle Link) là giao thức truyền thông nhẹ (lightweight protocol) dùng cho các hệ thống tự hành không người lái. Trong dự án của nhóm, MAVLink truyền qua UDP, backend dùng pymavlink để gửi/nhận các gói tin.

- **Flask** là một micro web framework của Python. Trong dự án thì Flask là cầu nối trung gian giữa web và PX4: Nhận yêu cầu từ frontend (ví dụ: tải waypoint, điều khiển mission); xử lý và chuyển đổi sang gói MAVLink để gửi cho PX4, đồng thời nhận telemetry từ PX4 qua MAVLink và truyền lại cho frontend theo thời gian thực.

2.2. Mô hình hệ thống / mô hình mạng của dự án

Hệ thống mô phỏng USV (Unmanned Surface Vehicle) được xây dựng theo kiến trúc 3 lớp được biểu diễn trong Hình 1.



Hình 1. Sơ đồ mô hình hệ thống

2.2.1. Lớp mô phỏng (PX4 Autopilot + SITL)

- PX4 SITL chạy trên máy tính, mô phỏng autopilot và môi trường bay.
- PX4 xuất dữ liệu MAVLink qua UDP trên cổng 14540 đến Backend (hoặc GCS).
- PX4 chịu trách nhiệm điều khiển bay tự động, thực thi mission waypoint do Frontend yêu cầu, và gửi telemetry/sensor data qua MAVLink (ví dụ: GPS, attitude, battery, state...).

2.2.2. Lớp Backend (Flask Server)

Backend đóng vai trò như một MAVLink Gateway, thực hiện:

- Kết nối với PX4 qua MAVLink với UDP port: 14540
- Nhận và gửi gói MAVLink qua UDP (pymavlink).
- Xử lý và phân tích thông điệp MAVLink.

- Cung cấp API REST cho frontend: Chọn waypoint - Upload mission - Start mission - Lấy GPS, tốc độ, battery.
- Điều phối trạng thái hệ thống thông qua biến điều khiển `polling_enabled` và `mavlink_lock`.

Điểm mới:

+ Cơ chế lock và polling control để tránh xung đột MAVLink khi vừa upload mission vừa đọc telemetry (tránh mất ACK).

* Giới thiệu các gói message có cú pháp như thế nào.

a. UPLOAD MISSION:

MISSION_CLEAR_ALL

- Xóa toàn bộ mission cũ trên autopilot trước khi upload nhiệm vụ mới.
- Gửi bằng `master.mav.mission_clear_all_send()`.
- Các tham số truyền vào:

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID

MISSION_COUNT

- Thông báo số lượng waypoint sắp gửi, giúp autopilot chuẩn bị bộ nhớ.
- Gửi bằng `master.mav.mission_count_send()`.
- Các tham số truyền vào:

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
count	uint16_t		Number of mission items in the sequence

MISSION_ITEM_INT

- Mã hóa waypoint thành gói MAVLink gửi tới PX4.
- Gửi bằng master.mav.mission_item_int_send().
- Các tham số truyền vào:

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
seq	uint16_t		Waypoint ID (sequence number). Bắt đầu với 0
frame	uint8_t	<u>MAV_FRAME</u>	The coordinate system of the waypoint.
command	uint16_t	<u>MAV_CMD</u>	The scheduled action for the waypoint.
current	uint8_t		false:0, true:1
autocontinue	uint8_t		Autocontinue
param1	float		PARAM1
param2	float		PARAM2
param3	float		PARAM3,
param4	float		PARAM4,
x	int32_t		PARAM5 / local: x position in meters * 1e4, global: latitude in degrees * 10 ⁷
y	int32_t		PARAM6 / y position: local: x

Field Name	Type	Values	Description
			position in meters * 1e4, global: longitude in degrees * 10 ⁷
z	float		PARAM7 / z position: global: altitude in meters

Trong đó:

MAV_CMD: sẽ dùng MAV_CMD_NAV_WAYPOINT để gửi waypoint.

x, y, z: là các tham số latitude, longitude và altitude.

b. GET MISSION

MISSION_REQUEST_LIST

- Yêu cầu danh sách toàn bộ waypoint từ PX4.
- Gửi bằng master.mav.mission_request_list_send().
- Các tham số truyền vào:

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID

MISSION_REQUEST_INT

- Yêu cầu waypoint theo thứ tự sequence.
- Gửi bằng master.mav.mission_request_int_send().
- Các tham số truyền vào:

Field Name	Type	Values	Description
------------	------	--------	-------------

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
seq	uint16_t		Sequence

c. Gửi lệnh START MISSION

Lệnh START MISSION được sử dụng để kích hoạt và bắt đầu thực thi các waypoint trong danh sách nhiệm vụ đã nạp vào autopilot.

Trong MAVLink, lệnh này được gửi thông qua hàm *master.mav.command_long_send()* với tham số MAV_CMD_MISSION_START, cho phép phương tiện chuyển sang chế độ bay theo mission đã cấu hình.

d. Telemetry nhận:

HEARTBEAT

- Thông báo trạng thái hệ thống, component.
- Các trường type, autopilot, component_id giúp backend xử lý thông điệp đúng.

MISSION_ACK

- Là message xác nhận trong giao thức MAVLink, được PX4 gửi về sau khi nhận và xử lý toàn bộ danh sách waypoint.
- Trường type cho biết kết quả: nếu type = 0 thì là xác nhận thành công, còn nếu type khác 0 thì báo lỗi.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID

Field Name	Type	Values	Description
type	uint8_t	<u>MAV_MISSION_RESULT</u>	Mission result.

GPS_RAW_INT

- Thông điệp GPS_RAW_INT cung cấp dữ liệu thô từ GPS, bao gồm tọa độ vĩ độ, kinh độ và độ cao. Đây là dữ liệu cơ bản mà autopilot sử dụng để định vị và điều hướng.
- Các tham số trả về:

Field Name	Type	Units	Description
lat	int32_t	degE7	Vĩ độ
lon	int32_t	degE7	Kinh độ
alt	int32_t	mm	Độ cao

BATTERY_STATUS

- Cập nhật trạng thái pin của bộ điều khiển bay cho GCS. Trong phạm vi hệ thống, chỉ trường *battery_remaining* của thông điệp BATTERY_STATUS được sử dụng.
- Các tham số trả về:

Field Name	Type	Units	Values	Description
id	uint8_t			Battery ID
battery_remaining	int8_t	%	invalid:-1	Remaining battery energy. Values: [0-100] hoặc -1

VFR_HUD

- Là một thông điệp MAVLink dùng để truyền các thông số điều hướng cơ bản của phương tiện. Đây là gói dữ liệu thường hiển thị trên HUD của GCS.
- Tham số trả về:

Field Name	Type	Units	Description
groundspeed	float	m/s	Current ground speed.
heading	int16_t	deg	Current heading in compass units (0-360, 0=north).

MISSION_CURRENT

- Thông báo số thứ tự waypoint hiện tại mà USV đang hướng tới/đang thực hiện. Dữ liệu này giúp GCS/backend theo dõi tiến độ mission, đảm bảo hệ thống điều khiển và giám sát real-time.
- Tham số trả về:

Field Name	Type	Values	Description
seq	uint16_t		Sequence
total ++	uint16_t	invalid:UINT16_MAX	Total number of mission items on vehicle
mission_state ++	uint8_t	invalid:0 <u>MISSION_STATE_UNKNOWN</u>	Mission state machine state. <u>MISSION_STATE_UNKNOWN</u> if state reporting not supported.

2.2.3. Lớp Frontend (Web Browser)

Gồm 3 giao diện chính:

- index.html: Hiển thị bản đồ tương tác (Leaflet, OpenStreetMap) cho phép người dùng: Click chọn waypoint - Gửi mission lên backend - Theo dõi vị trí thời gian thực.
- dashboard.html: Hiển thị các thông số: Battery, tốc độ, heading.
- stream.html: phát video từ camera.

Điểm mới: Tạo web interface hỗ trợ tương tác người dùng trong quá trình thực hiện đề tài USV giám sát thay vì chỉ điều khiển drone thuần túy với PX4 Autopilot.

2.2. Nhiệm vụ của các khối / nút mạng

Bảng 1. Nhiệm vụ của các khối mạng

Thành phần	Vai trò - nhiệm vụ
PX4 SITL	Sinh dữ liệu mô phỏng, gửi MAVLink. qua UDP; điều khiển mission.
MAVLink Protocol	Giao thức truyền thông giữa PX4 và Backend; gồm telemetry, command, mission protocol.
Flask Backend	Cổng giao tiếp trung tâm; phân tích MAVLink; cung cấp REST API.
Mission Module	Upload waypoints, gửi MISSION_ACK, start mission.
Frontend	Tương tác người dùng, gửi yêu cầu, hiển thị bản đồ và trạng thái USV.
Map Engine	Hiển thị bản đồ thật từ OSM, chọn waypoint.
Client	Tạo yêu cầu tới backend.

2.3. Mô tả kịch bản mô phỏng

Bước 1: Khởi động mô phỏng

- Khởi chạy PX4 SITL (gazebo)
- PX4 tự động phát luồng MAVLink qua UDP tại địa chỉ 127.0.0.1:14540
- Backend mở socket UDP, kết nối đến PX4 và chờ nhận HEARTBEAT để xác nhận liên kết MAVLink đã thiết lập thành công.

Bước 2: Người dùng lập kế hoạch mission

- Người dùng tương tác với bản đồ bằng cách click để chọn các điểm waypoint.
- Các điểm được tự động thêm vào danh sách mission và hiển thị trực quan trên bản đồ.

- Sau khi hoàn tất, người dùng nhấn nút "Upload" để gửi danh sách waypoint lên backend. Sau đó nhấn nút "Start" để bắt đầu hành trình.

Bước 3: Backend xử lý upload mission

Backend thực hiện quy trình upload mission với cơ chế đồng bộ hóa:

- Dừng toàn bộ API polling bằng cách thiết lập `polling_enabled = false`
- Gửi lệnh `MISSION_CLEAR_ALL` để xóa mission cũ trên PX4
- Gửi `MISSION_COUNT` thông báo số lượng waypoint
- Gửi tuần tự từng `MISSION_ITEM_INT` với thông tin tọa độ waypoint
- Chờ `MISSION_ACK` từ PX4 để xác nhận upload thành công
- Mở lại polling (`polling_enabled = true`) sau khi hoàn tất

Điểm mới: Cơ chế tạm dừng polling và sử dụng lock để tránh xung đột khi nhiều API cùng đọc/ghi trên một luồng UDP, đảm bảo reliability của quá trình upload mission.

Bước 4: Khởi chạy mission

- Frontend gọi API start mission
- Backend gửi lệnh `MAV_CMD_MISSION_START` đến PX4
- PX4 bắt đầu điều khiển USV di chuyển tự động theo các waypoint đã thiết lập

Bước 5: Theo dõi và giám sát real-time

Trong quá trình thực thi mission, hệ thống cung cấp giám sát real-time:

Dữ liệu từ backend:

- Vị trí GPS (`GLOBAL_POSITION_INT`)
- Tốc độ và heading (`VFR_HUD`)
- Trạng thái pin (`BATTERY_STATUS`)

Frontend cập nhật:

- Vị trí thuyền trên bản đồ theo thời gian thực
- Dashboard hiển thị các thông số telemetry
- Tiến độ mission và trạng thái hoạt động

Bước 6: Kết thúc mission

- PX4 báo `MISSION_CURRENT == MISSION_TOTAL`, nhiệm vụ hoàn tất.
- Hệ thống thông báo hoàn thành mission cho người dùng

Điểm mới trong kịch bản mô phỏng:

- Cơ chế đồng bộ hóa đảm bảo độ tin cậy khi upload mission
- Giao diện trực quan cho phép người dùng dễ dàng lập kế hoạch và giám sát
- Xử lý real-time cả dữ liệu điều hướng và môi trường trên cùng một platform

Kịch bản xử lý lỗi:

- **Mất kết nối với PX4:**
 - + Backend phát hiện mất kết nối (không nhận được heartbeat) và thông báo lỗi cho frontend.
 - + Backend thử kết nối lại sau một khoảng thời gian.
- **Lỗi khi upload mission:** Nếu không nhận được MISSION_ACK từ PX4, backend thông báo lỗi và cho phép thử lại.

2.4. Điều khiển waypoint tự động (Mission Control) - Điểm mới của nhóm

Mục tiêu:

- Ở frontend chọn waypoint trên bản đồ, gửi về backend.
- Backend chuyển waypoint thành gói MAVLink (chuyển đổi lat/lon thành scaled int) gửi PX4.
- PX4 nhận, tự động lái phương tiện đi theo các waypoint.
- Backend theo dõi tiến độ mission, báo lại cho frontend.

Mission control đã được thử nghiệm với SITL và hoạt động ổn định với ACK.

3. Kết quả và đánh giá hệ thống

3.1. Các metrics sử dụng để đánh giá hệ thống

- + Độ trễ phản hồi lệnh (RTT MAVLink): đo thời gian từ khi backend gửi lệnh MAVLink đến PX4 và nhận ACK (MISSION_ACK).
- + Latency: đo độ trễ giữa frontend gửi request - backend xử lý - đọc MAVLink - trả kết quả về. Các endpoint được đo gồm mission-progress, vehicle-info, vehicle-position.
- + Tỷ lệ mất gói (Packet Loss): dùng đánh giá độ tin cậy của kênh telemetry và progress khi nhiều API cùng truy cập.
- + Độ đúng đắn dữ liệu: kiểm tra mức độ khớp giữa dữ liệu MAVLink (ví dụ: vị trí GPS, trạng thái mission) và dữ liệu trong file log ghi lại từ quá trình mô phỏng. Việc đối chiếu này giúp xác nhận tính chính xác của backend trong việc tiếp nhận, xử lý và cung cấp dữ liệu cho hệ thống.

3.2. Mô tả các file log kết quả.

Trong quá trình vận hành, hệ thống ghi lại các thông số về truyền thông mạng (latency, loss, RTT - Round Trip Time) tại thư mục EPIC_B/logs. Các thông số này phục vụ mục đích đánh giá mức độ ổn định, độ trễ và độ tin cậy của kênh truyền MAVLink qua UDP.

3.2.1. File “command_rtt.csv” - Độ trễ phản hồi lệnh (Round Trip Time)

Ý nghĩa:

- File này ghi lại thời gian thực hiện vòng lặp gửi lệnh từ backend - PX4 - PX4 phản hồi lại (MISSION_ACK). Đây là thông số quan trọng để đánh giá tốc độ phản hồi của kênh điều khiển

Kết quả ghi được:

	A	B	C	D
1	21:48.6	MISSION_ACK	2.242258	TRUE

Hình 2. Kết quả đo RTT của MISSION_ACK

Nhận xét:

- Tất cả yêu cầu đều nhận được phản hồi (success = True)
- RTT rất nhỏ, đạt giá trị tốt khi chạy trong môi trường local.
- Thời gian RTT trung bình ≈ 2.24 ms; không có mất gói. Độ trễ 2.24 ms phản ánh kênh điều khiển ổn định, không có hiện tượng nghẽn mạng, đáp ứng được điều kiện cho các ứng dụng điều khiển thời gian thực.

Kết luận:

Kênh điều khiển hoạt động nhanh và ổn định với giá trị RTT rất nhỏ, đáp ứng tốt yêu cầu của hệ thống điều khiển USV.

3.2.2. File “mission-progress_latency.csv” - Độ trễ tiến trình nhiệm vụ

Ý nghĩa:

- File ghi lại độ trễ khi backend yêu cầu PX4 gửi thông tin tiến độ nhiệm vụ.

Kết quả ghi được:

	A	B	C	D	E
1	timestamp	endpoint	latency_ms	status	loss_ratio
2	21:42.4	mission-progress	604.846076	success	0
3	21:43.4	mission-progress	1003.39052	success	0
4	21:44.5	mission-progress	623.969423	success	0
5	21:45.5	mission-progress	486.673719	success	0
6	21:46.5	mission-progress	499.0949	success	0
7	21:48.5	mission-progress	668.0379	success	0
8	21:49.6	mission-progress	718.789929	success	0
9	21:52.7	mission-progress	717.15022	success	0
10	21:53.7	mission-progress	622.690443	success	0
11	21:56.7	mission-progress	737.862698	success	0

Hình 3. Kết quả đo độ trễ tiến trình nhiệm vụ

Nhận xét

- Latency dao động khoảng 500 ms đến khoảng 1000 ms
- Không có mất gói (loss_ratio = 0.0)
- Đây là kết quả hợp lý, không phải lỗi mạng.
- Tỷ lệ mất gói 0% là rất tốt, đạt độ tin cậy.
- Nguyên nhân latency dao động khá lớn do PX4 chỉ gửi thông tin mission-progress theo chu kỳ 1 Hz. Backend chạy liên tục, lúc nhận ngay (≈ 500 ms), lúc đợi chu kỳ tiếp theo (≈ 1000 ms).

Kết luận:

Độ trễ mission-progress dao động 500–1000 ms do đặc tính chu kỳ của PX4. Hệ thống hoạt động đúng thiết kế, không mất gói và đảm bảo độ tin cậy.

3.2.3. File “vehicle-info_latency.csv” - Độ trễ thông tin trạng thái phương tiện

Ý nghĩa:

- Backend sẽ yêu cầu PX4 gửi trạng thái (vehicle-info) như: battery, heading, speed.

Kết quả đo được:

	A	B	C	D	E
1	timestamp	endpoint	latency_ms	status	loss_ratio
2	21:41.8	vehicle-info	809.8653	success	0
3	21:43.8	vehicle-info	364.7837	success	0
4	21:45.8	vehicle-info	365.4234	success	0
5	21:47.9	vehicle-info	886.036	success	0
6	21:51.9	vehicle-info	1957.601	success	0
7	21:56.0	vehicle-info	1901.508	success	0

Hình 4. Kết quả đo độ trễ thông tin và trạng thái phương tiện

Nhận xét:

- Latency dao động từ khoảng 350 ms đến 2000 ms.
- Tuy dao động lớn nhưng vẫn hợp lý với loại dữ liệu này.
- Không có mất mát gói, hệ thống đáng tin cậy.
- Nguyên nhân latency khá cao, dao động lớn vì:
 - + Đọc nhiều loại dữ liệu với các message khác nhau.
 - + Đây là request/response không chạy theo tần số cố định.
 - + PX4 cần dựng message mới để phản hồi, mất thời gian hơn.

Kết luận:

Vehicle-info có độ trễ 350 - 2000 ms, phù hợp với kiểu truyền thông yêu cầu phản hồi và không ảnh hưởng đến điều khiển thời gian thực.

3.2.4. File “vehicle-position_latency.csv” - Độ trễ dữ liệu vị trí

Ý nghĩa:

- Đo latency khi PX4 stream vị trí về backend để nhận vị trí thời gian thực của thuyền.

Kết quả ghi được:

	A	B	C	D	E
1	timestamp	endpoint	latency_ms	status	loss_ratio
2	21:42.4	vehicle-position	16.944369	success	0
3	21:43.4	vehicle-position	18.036323	success	0
4	21:43.5	vehicle-position	18.795005	success	0
5	21:44.5	vehicle-position	18.159813	success	0
6	21:45.5	vehicle-position	2.223796	success	0
7	21:46.0	vehicle-position	14.864929	success	0
8	21:46.5	vehicle-position	5.41071	success	0
9	21:46.5	vehicle-position	20.801877	success	0

Hình 5. Kết quả đo độ trễ dữ liệu vị trí

Nhận xét

- Latency thấp, dao động từ khoảng 2 đến 20 ms.
- Không có mất gói.
- Giải thích:
 - + Vị trí được PX4 gửi theo tần số cao (10 - 20 Hz).
 - + UDP local nên độ trễ gần như không đáng kể.
- Đây là kênh quan trọng đối với quá trình điều khiển USV.
- Kết quả đạt yêu cầu thời gian thực, ổn định và tin cậy.

Kết luận:

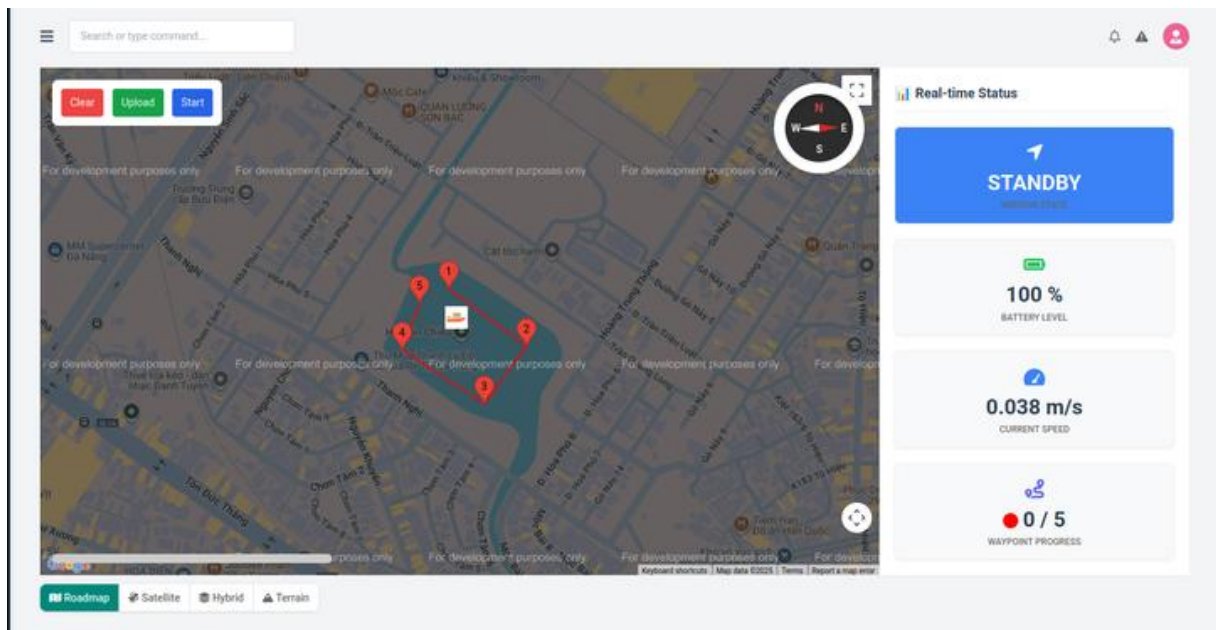
Vehicle-position có latency khoảng 2 - 20 ms, đảm bảo yêu cầu của hệ thống định vị thời gian thực và hỗ trợ điều khiển chính xác.

3.3. Kết quả chạy thực nghiệm và so sánh

3.3.1. Kết quả đo khi thuyền ở vị trí home khởi chạy từ PX4

Đầu tiên ta chạy khởi tạo các lệnh để bắt đầu chương trình, USV ban đầu đang đứng yên với tọa độ set home tại hồ Liên Chiểu, TP Đà Nẵng.

Thực hiện chọn các waypoint cho hành trình.



Hình 6. Vị trí USV và waypoints tại vị trí home

	timestamp	lat	lon	alt	speed	heading	battery	mission_current	mission_total
12	153186000	16.0661878	108.1611129	2.04	0.025843484327197075	88	100	0	5
13	153186000	16.0661878	108.1611129	2.04	0.027536489069461823	88	100	0	5
14	155860000	16.0661881	108.1611128	2.051	0.027536489069461823	88	100	0	5
15	156190000	16.0661879	108.1611134	2.087	0.027536489069461823	88	100	0	5
16	156850000	16.0661879	108.161113	2.078	0.027536489069461823	88	100	0	5
17	157212000	16.0661881	108.161113	2.057	0.027536489069461823	88	100	0	5
18	157212000	16.0661881	108.161113	2.057	0.027536489069461823	88	100	0	5
19	157212000	16.0661881	108.161113	2.057	0.027536489069461823	88	100	0	5
20	157212000	16.0661881	108.161113	2.057	0.027536489069461823	88	100	0	5
21	157212000	16.0661881	108.161113	2.057	0.03787766396999359	88	100	0	5
22	159786000	16.0661882	108.1611131	2.101	0.03787766396999359	88	100	0	5
23	160248000	16.0661881	108.1611133	2.046	0.03787766396999359	88	100	0	5

Hình 7. Kết quả bảng log backend MAVLink (home)

Từ bảng log này, lần lượt ta sẽ so sánh với dữ liệu có được từ PX4

- GPS: lấy lat, lon, alt, speed, heading, thời gian cập nhật

```

TOPIC: sensor_gps
sensor_gps
  timestamp: 159952000 (0.012000 seconds ago)
  timestamp_sample: 159952000 (0 us before timestamp)
  latitude_deg: 16.066188
  longitude_deg: 108.161113
  altitude_msl_m: 2.034106
  altitude_ellipsoid_m: 2.034106
  time_utc_usec: 0
  device_id: 11469068 (Type: 0xAF, SIMULATION:1 (0x01))
  s_variance_m_s: 0.40000
  c_variance_rad: 0.10000
  eph: 0.90000
  epv: 1.78000
  hdop: 0.70000
  vdop: 1.10000
  noise_per_ms: 0
  jamming_indicator: 0
  vel_m_s: 0.01882
  vel_n_m_s: 0.00574
  vel_e_m_s: -0.01793
  vel_d_m_s: 0.00708
  cog_rad: -1.26086
  heading: 1.54852
  heading_var: 0.01171
  unaided_heading: 0.00556
  delta_heading: 1.51882

```

Hình 8. Dữ liệu GPS từ PX4 (home)

Bảng 2. Bảng so sánh dữ liệu GPS giữa backend MAVLink và PX4 (tại home)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Latitude	16.0661878°	16.066188°	Tốt, sai số nhỏ
Longitude	108.161129°	108.161113°	Tốt, sai số nhỏ
Altitude	2.04m – 2.078m	2.034106m	Nằm trong khoảng chấp nhận
Speed	~ 0 m/s	~ 0 m/s	Nằm trong khoảng chấp nhận
Heading	88°	1,54 rad ~ 88.28°	Chênh lệch rất nhỏ

- Battery: lấy remaining, thời gian cập nhật

```

TOPIC: battery_status
battery_status
timestamp: 159964000 (0.002000 seconds ago)
voltage_v: 16.20000
current_a: -1.00000
current_average_a: 15.00000
discharged_mah: -43.18502
remaining: 0.99995

```

Hình 9. Dữ liệu Battery từ PX4 (home)

Bảng 3. Bảng so sánh dữ liệu Battery giữa backend MAVLink và PX4 (tại home)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Battery	100%	Remaining: 0.99995	Dữ liệu chính xác cao, sai số rất nhỏ

- MISSION: lấy seq_curent và seq_total

```

TOPIC: mission_result
mission_result
timestamp: 5740000 (154.623993 seconds ago)
mission_id: 3119525417
geofence_id: 0
home_position_counter: 117
seq_reached: -1
seq_current: 0
seq_total: 5

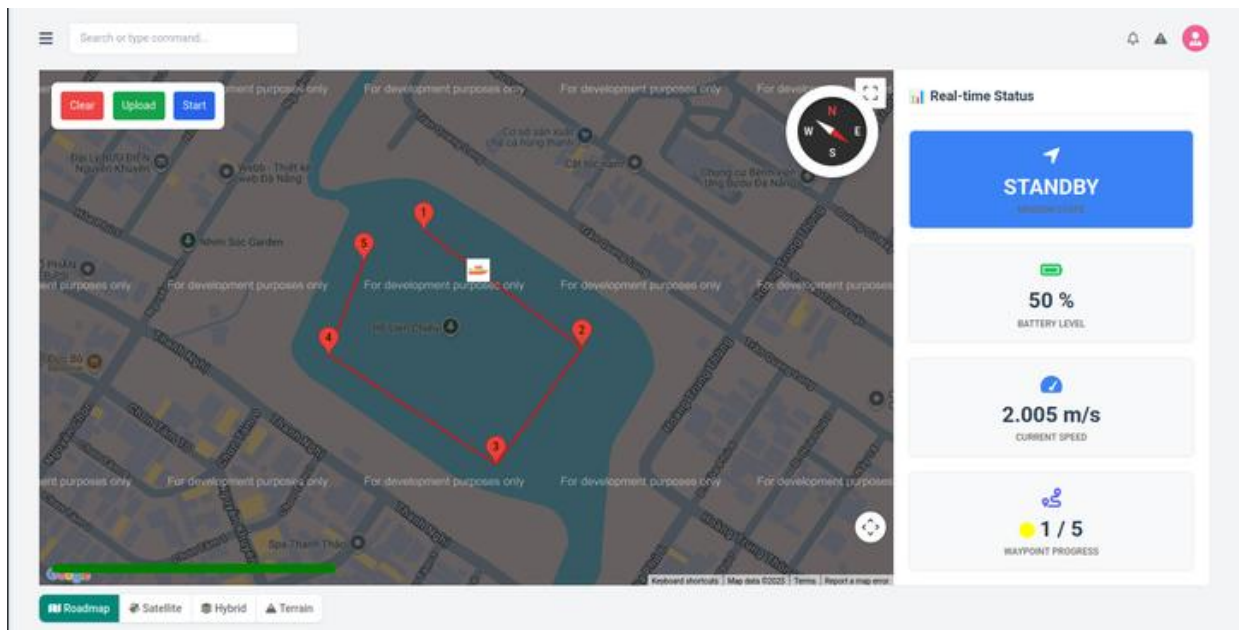
```

Hình 10. Dữ liệu Mission từ PX4 (home)

Bảng 4. Bảng so sánh dữ liệu Mission giữa backend MAVLink và PX4 (tại home)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Mission hiện tại	0	seq_curent: 0	Chính xác
Tổng mission	5	seq_total: 5	Chính xác

3.3.2. Kết quả khi thuyền di chuyển theo mission



Hình 11. Vị trí USV và waypoints khi đi chuyển

	timestamp	lat	lon	alt	speed	heading	battery	mission_current	mission_total
130	76264000	16.0663999	108.1612747	2.109	1.9681748151779175	125	50	1	5
131	77088000	16.0663914	108.1612868	2.069	1.9699338674545288	125	50	1	5
132	77518000	16.0663866	108.1612932	2.037	1.9699338674545288	125	50	1	5
133	78046000	16.0663809	108.1613011	2.112	1.9699338674545288	125	50	1	5
134	78276000	16.0663784	108.1613044	2.098	1.9699338674545288	125	50	1	5
135	79036000	16.0663703	108.161316	2.128	1.9736127853393555	125	50	1	5
136	79530000	16.0663648	108.1613229	2.079	1.9736127853393555	125	50	1	5
137	80026000	16.0663593	108.1613305	2.077	1.9736127853393555	125	50	1	5
138	80290000	16.0663567	108.1613348	2.071	1.9736127853393555	125	50	1	5
139	80752000	16.0663517	108.1613416	2.028	2.0047528743743896	125	50	1	5
140	81478000	16.0663441	108.1613526	2.037	2.0047528743743896	125	50	1	5

Hình 12. Kết quả bảng log backend MAVLink (đi chuyển)

Từ bảng log này, lần lượt ta sẽ so sánh với dữ liệu có được từ PX4

- GPS: lấy lat, lon, alt, speed, heading, thời gian cập nhật

```

TOPIC: sensor_gps
sensor_gps
timestamp: 79234000 (0.028000 seconds ago)
timestamp_sample: 79234000 (0 us before timestamp)
latitude_deg: 16.066368
longitude_deg: 108.161319
altitude_msl_m: 2.101035
altitude_ellipsoid_m: 2.101035
time_utc_usec: 0
device_id: 11469068 (Type: 0xAF, SIMULATION:1 (0x01))
s_variance_m_s: 0.40000
c_variance_rad: 0.10000
eph: 0.90000
epv: 1.78000
hdop: 0.70000
vdop: 1.10000
noise_per_ms: 0
jamming_indicator: 0
vel_m_s: 1.95890
vel_n_m_s: -1.15639
vel_e_m_s: 1.58115
vel_d_m_s: -0.03962

heading: 2.19070
heading_var: 0.00076
unaided_heading: 0.62131
delta_heading: 1.51902

```

Hình 13. Dữ liệu GPS từ PX4 (đi chuyển)

Bảng 5. Bảng so sánh dữ liệu GPS giữa backend MAVLink và PX4 (đi chuyển)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Latitude	16.066399°	16.066416°	Tốt, sai số nhỏ
Longitude	108.1612747°	108.161253°	Tốt, sai số nhỏ
Altitude	2.037m - 2.128m	2.062613m	Nằm trong khoảng chấp nhận
Speed	1.968 m/s – 2.004 m/s	1.9589 m/s	Nằm trong khoảng chấp nhận
Heading	125°	2.19 rad ~ 125.54°	Chênh lệch rất nhỏ

- Battery: lấy remaining, thời gian cập nhật

```

TOPIC: battery_status
battery_status
timestamp: 79260000 (0.002000 seconds ago)
voltage_v: 15.30000
current_a: -1.00000
current_average_a: 4.96771
discharged_mah: -20.77827
remaining: 0.50003

```

Hình 14. Dữ liệu Battery từ PX4 (di chuyển)

Bảng 6. Bảng so sánh dữ liệu Battery giữa backend MAVLink và PX4 (di chuyển)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Battery	50%	Remaining: 0.50003	Dữ liệu chính xác cao

- MISSION: lấy seq_curent và seq_total

```

TOPIC: mission_result
mission_result
timestamp: 52256000 (27.006001 seconds ago)
mission_id: 3119525417
geofence_id: 0
home_position_counter: 121
seq_reached: 0
seq_current: 1
seq_total: 5

```

Hình 15. Dữ liệu Mission từ PX4 (di chuyển)

Bảng 7. Bảng so sánh dữ liệu Mission giữa backend MAVLink và PX4 (di chuyển)

Thông số	BACKEND MAVLINK	PX4 DATA COMMAND	Nhận xét
Mission hiện tại	1	seq_curent: 1	Chính xác
Tổng mission	5	seq_total: 5	Chính xác

4. Kết luận

Dự án đã xây dựng thành công việc mô phỏng hành trình USV dựa trên PX4 SITL kết hợp Gazebo, thông qua giao thức MAVLink truyền qua UDP. Chương trình thực hiện ổn định quá trình trao đổi dữ liệu giữa PX4 và backend, từ gửi waypoint, nhận trạng thái, đến điều khiển hành trình tự động.

Bên cạnh đó, việc thiết kế giao diện lập kế hoạch mission trực quan giúp người dùng dễ dàng tạo, tải waypoints và theo dõi lộ trình. Cơ chế đồng bộ hóa giao tiếp (lock và tạm dừng polling) đảm bảo việc upload mission diễn ra chính xác và tin cậy.

Kết quả đạt được đảm bảo việc truyền dữ liệu thời gian thực qua MAVLink/UDP với các gói tin ổn định, tin cậy và phản hồi kịp thời.