

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



BÁO CÁO HỌC PHẦN LẬP TRÌNH MẠNG

**ĐỀ TÀI:
ĐIỀU KHIỂN DRONE THEO QUÝ ĐẠO**

Giảng viên hướng dẫn : TS. Nguyễn Văn Hiếu

Sinh viên thực hiện :

Lê Thanh Bản - Lớp: 22KTMT1 - MSSV: 106220209

Lưu Văn Thành Huy - Lớp: 22KTMT1 - MSSV: 106220218

Trương Văn Tú - Lớp: 22KTMT1 - MSSV: 106220240

Đà Nẵng, tháng 11/2025

MỤC LỤC

1. Mô hình hệ thống/mạng và chức năng các khối /nút mạng:	4
1.1. Tổng quan kiến trúc:	4
1.2. Chức năng chi tiết của các khối:	4
1.2.1. Khối điều khiển người dùng (User Control):	4
1.2.2. Khối điều khiển trung tâm / bộ não hệ thống (Offboard Control):	4
1.2.3. Khối mô phỏng phần cứng / môi trường (PX4 SITL + Gazebo):	5
1.2.4. Khối hiển thị trực quan (Visualizer/RViz):	5
2. Mô tả kịch bản mô phỏng:	5
3. Phân tích kết quả:	6
3.1. Các metrics/ thông số dùng để đánh giá hệ thống và phân tích các thông số metrics (tính hợp lý , ổn định và đáp ứng yêu cầu tiêu chuẩn):	6
3.1.1. Các thông số /metrics đánh giá độ chính xác điều khiển (Control Accuracy):	7
3.1.2. Độ ổn định bay (Flight Stability):	12
3.1.3. Mức độ an toàn hệ thống (System Safety):	15
3.2. Mô tả các files kết quả lưu trữ thông tin:	17
3.2.1. File log chính của PX4:	18
3.2.2. Logged Messages (Nhật ký sự kiện):	18
3.2.3. Non-default Parameters (Các tham số cấu hình thay đổi):	19
3.2.4. Performance Counters (Thống kê hiệu năng hệ thống):	20
3.2.5. Flight Review Charts (Biểu đồ phân tích dữ liệu):	21
3.3. Kết quả demo:	21
3.5. Kết luận chung về cơ chế điều khiển trong PX4 Offboard:	35
Tài liệu tham khảo	36

PHÂN CÔNG NHIỆM VỤ

STT	HỌ VÀ TÊN	NHIỆM VỤ	KHÓI LUỢNG
11	Lưu Văn Thành Huy	Xây dựng khối điều khiển trung tâm (điều khiển thủ công hoặc quỹ đạo Sine)	33%
3	Lê Thanh Bản	Xây dựng khối điều khiển người dùng và khối điều khiển trung tâm (quỹ đạo tròn)	33%
32	Trương Văn Tú	Xây dựng khối hiển thị trực quan.	33%

1. Mô hình hệ thống/mạng và chức năng các khối /nút mạng:

1.1. Tổng quan kiến trúc:

Hệ thống điều khiển UAV trong mô phỏng được thiết kế theo kiến trúc nhiều khối gồm bốn khối chính gồm:

- Khối điều khiển người dùng (User Control Node)
- Khối điều khiển trung tâm/ bộ não hệ thống (Offboard Control Node)
- Khối mô phỏng phần cứng/ môi trường (PX4 SITL + Gazebo)
- Khối hiển thị trực quan (Visualizer/ RViz)

Trong đó: PX4 dùng khung tọa độ NED (x North, y East, z Down); khi hiển thị trên RViz/ROS (theo chuẩn ENU) cần đổi dấu tương ứng hoặc thực hiện chuyển đổi frame. Khối điều khiển trung tâm phải gửi giá trị điều khiển (setpoints) liên tục (PX4 yêu cầu ≥ 2 Hz); code hiện gửi ở 50 Hz (0.02 s) để đảm bảo mượt.

1.2. Chức năng chi tiết của các khối:

1.2.1. Khối điều khiển người dùng (User Control):

Chức năng:

- Đọc phím từ terminal ở raw mode.
- Chuyển phím thành lệnh vận tốc (Twist), lệnh khởi động arm/disarm (Bool) và lệnh chọn chế độ quỹ đạo (String).
- Gửi các lệnh này đến trung tâm điều khiển.

Điểm mới: xử lý phím thời gian thực bằng chế độ "raw mode", đảm bảo độ trễ thấp và không cần nhấn Enter.

1.2.2. Khối điều khiển trung tâm / bộ não hệ thống (Offboard Control):

Chức năng:

- Nhận lệnh từ User Control Node.
- Quản lý trạng thái UAV (khởi động, cất cánh, chế độ điều khiển, hạ cánh).
- Triển khai tập hợp các bước hoạt động của UAV – state machine (IDLE (chờ lệnh) → ARMING (khởi động) → TAKEOFF (cất cánh) → LOITER (giữ vị trí tạm thời) → OFFBOARD (nhận tín hiệu điều khiển)).
- Tính toán vận tốc hoặc quỹ đạo phù hợp với từng kịch bản bay.
- Truyền lệnh chính xác sang PX4 SITL theo giao thức MAVROS/MAVLink.

Điểm mới:

- Tích hợp State Machine tự động chuyển trạng thái: DISARMED → ARMED → TAKEOFF → OFFBOARD.

- Hỗ trợ hai chế độ: manual (điều khiển tay) và auto (bay theo quỹ đạo định sẵn).

1.2.3. Khối mô phỏng phần cứng / môi trường (PX4 SITL + Gazebo):

Chức năng:

- Mô phỏng UAV, động cơ, cảm biến, khí động học.
- Nhận lệnh điều khiển từ khói điều khiển trung tâm (Offboard Control).
- Gửi dữ liệu phản hồi (vị trí, vận tốc, trạng thái góc) về ROS2 để hiển thị.

1.2.4. Khối hiển thị trực quan (Visualizer/RViz):

Chức năng:

- Hiển thị vị trí, hướng bay, độ cao, quỹ đạo chuyển động (trajectory) của UAV theo thời gian thực.
- Lưu lại quỹ đạo để đánh giá độ mượt và độ chính xác của thuật toán.

2. Mô tả kịch bản mô phỏng:

- Các tham số tiêu chuẩn dùng trong mô phỏng:

- A (amplitude SINE) = 5.0 m
- f (tần số SINE) = 0.1 Hz (chu kỳ 10 s)
- v_f (vận tốc tiến) = 2.0 m/s
- R (bán kính CIRCLE) = 10.0 m
- ω (omega, angular speed) = 0.1 rad/s
- z_{target} = 30 m (NED z = -30.0)
- publish_rate = 50 Hz (period = 0.02 s) – Tần số Offboard Control gửi gói lệnh điều khiển vị trí / vận tốc / hướng muôn drone đạt được (TrajectorySetpoint).

- Kịch bản 1: Điều khiển thủ công bằng phím (Manual Mode)

Mô tả:

- Người dùng điều khiển UAV hoàn toàn bằng bàn phím: tiến, lùi, rẽ trái/phải, tăng/giảm độ cao và xoay yaw.
- Mỗi phím tương ứng với một vector vận tốc theo trục X-Y-Z.
- Space để bật/tắt arm (khởi động), R để reset về điều khiển thủ công (manual).

Điểm mới:

- Điều khiển real-time, phản hồi ngay lập tức nhờ chế độ đầu vào thô (raw keyboard input); chế độ tăng tốc tích lũy (press-to-increase) được sử dụng trong điều khiển bàn phím (teleop).

- Kịch bản 2: Bay theo quỹ đạo hình SIN (Sine Trajectory)

Mô tả:

- UAV tự động đạt độ cao 30 m sau khi khởi động và cất cánh (arming & takeoff).
- Khi đạt độ cao, UAV di chuyển về phía trước với tốc độ không đổi $v_x = v_f$.
- Đồng thời dao động ngang theo hàm SIN theo thời gian:

$$y(t) = A \cdot \sin(\omega t)$$

- Vì hệ gửi các giá trị tốc độ cần đạt tới (velocity setpoints), ta sử dụng đạo hàm:

$$vy(t) = \frac{d}{dx} y(t) = A\omega \cos(\omega t)$$

với $\omega = 2\pi f$ nếu ta dùng f theo Hz

Điểm mới: Tính toán vận tốc theo thời gian thực từ quỹ đạo tham chiếu

- Kịch bản 3: Bay vòng tròn (Circle Trajectory)

Mô tả:

- UAV tự động tăng độ cao lên 30 m giống kịch bản 2.
- Sau đó UAV bay vòng tròn với bán kính cố định R theo phương trình tham chiếu:

$$x(t) = R \cos(\omega t), \quad y(t) = R \sin(\omega t)$$

- Vận tốc điều khiển cần gửi (velocity-based):

$$Vx(t) = -R\omega \sin(\omega t), \quad Vy(t) = R\omega \cos(\omega t).$$

Điểm mới:

- Sử dụng điều khiển theo vận tốc để sinh chuyển động tròn liên tục, giúp chuyển động mượt mà thay vì bay theo tập hợp điểm cố định (waypoint) gây giật, mất ổn định.

3. Phân tích kết quả:

3.1. Các metrics/ thông số dùng để đánh giá hệ thống và phân tích các thông số metrics (tính hợp lý, ổn định và đáp ứng yêu cầu tiêu chuẩn):

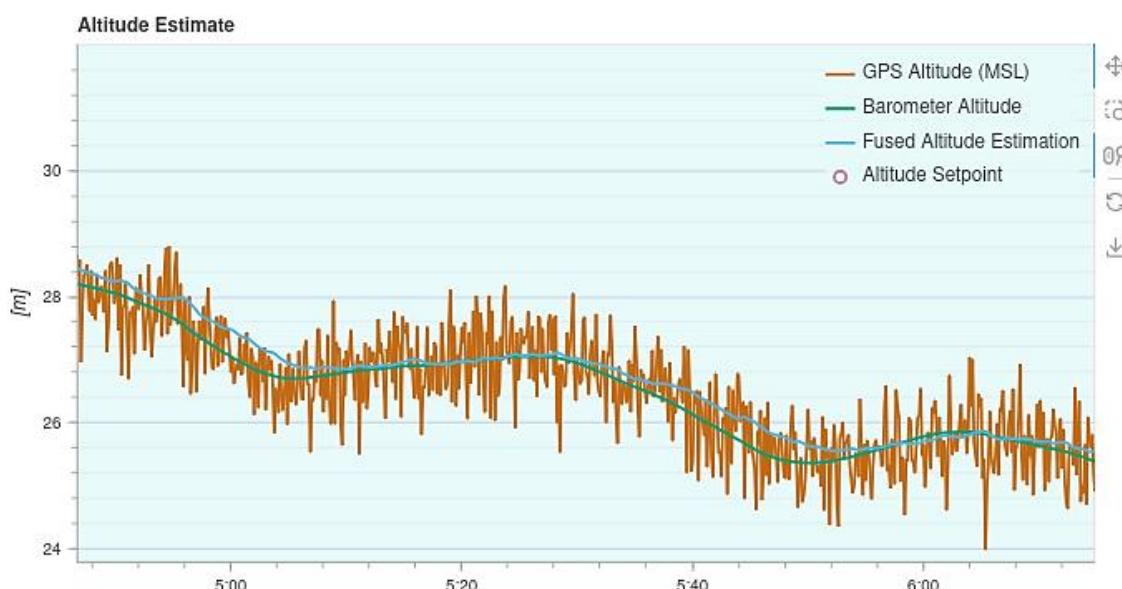
Metrics là thông số hoạt động gồm có vị trí, độ ổn định góc (yaw), tốc độ đạt được, mức rung và thời gian phản hồi,... Trong hệ thống mô phỏng điều khiển drone sử dụng ROS2 - PX4 Offboard Mode, các thông số đánh giá (metrics) đóng vai trò quan trọng trong việc xác định mức độ chính xác, ổn định và an toàn của hệ thống điều khiển. Mục tiêu phần này tập trung vào ba khía cạnh chính :

- Độ chính xác điều khiển (Control Accuracy)
- Độ ổn định bay (Flight Stability)
- Mức độ an toàn hệ thống (System Safety)

Các đánh giá được dựa trên dữ liệu log thu được từ PX4(.ulg) và quan sát setpoint so sánh với giá trị thực tế của drone trong quá trình bay mô phỏng SITL.

3.1.1. Các thông số /metrics đánh giá độ chính xác điều khiển (Control Accuracy):

a) Điều khiển độ cao (Altitude Tracking):



Biểu đồ thể hiện sự thay đổi độ cao của drone theo thời gian, bao gồm :

- Độ cao GPS (GPS Altitude – màu cam)
- Độ cao Barometer (Barometer Altitude – màu xanh lá)
- Độ cao ước lượng hợp nhất EKF (Fused Altitude – màu xanh dương)
- Giá trị độ cao đặt trước (Altitude Setpoint – chấm tím)

Từ đó cho thấy khả năng kiểm soát độ cao của hệ thống trong quá trình bay.

Những thông số/metrics đánh giá là:

- Altitude Setpoint : nó thể hiện giá trị độ cao mà hệ thống cần đạt từ đó thể hiện mốc tham chiếu để đánh giá chính xác điều khiển.
- GPS Altitude : nó thể hiện độ cao từ GPS từ đó đánh giá được mức nhiễu và độ tin cậy của GPS.
- Barometer Altitude : nó thể hiện độ cao đo từ cảm biến áp suất , từ đó kiểm tra độ ổn định đo lường.

- Fused Altitude (EKF) : nó thể hiện độ cao đã được hệ thống xử lý và kết hợp từ nhiều cảm biến để cho ra độ cao tin cậy nhất, từ đó đánh giá hiệu quả của lọc EKF và chất lượng điều khiển.

Đánh giá kết quả từ biểu đồ:

- Đánh giá về hệ thống duy trì độ cao ổn định: Giá trị Fused Altitude (EKF) và Barometer Altitude bám sát nhau, độ cao thay đổi mượt, không có dao động đột ngột, không xuất hiện “nhảy số”.

=> Chứng tỏ hệ thống có khả năng ổn định tốt, cảm biến hoạt động bình thường.

- Đánh giá về GPS: Đường GPS Altitude có biên độ nhiễu lớn, dao động liên tục. Tuy nhiên, đường Fused Altitude vẫn mượt và ổn định.

=> Chứng tỏ EKF hoạt động tốt, lọc nhiễu và kết hợp dữ liệu cảm biến hiệu quả.

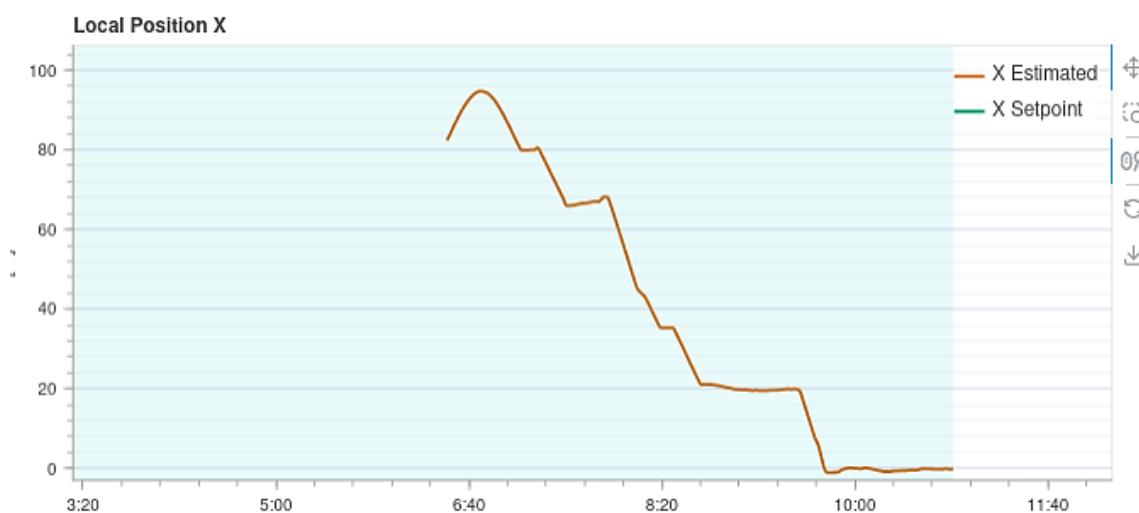
- Đánh giá về mức độ kiểm soát độ cao: Không có hiện tượng tự động tụt độ cao đột ngột và tăng độ cao bất thường.

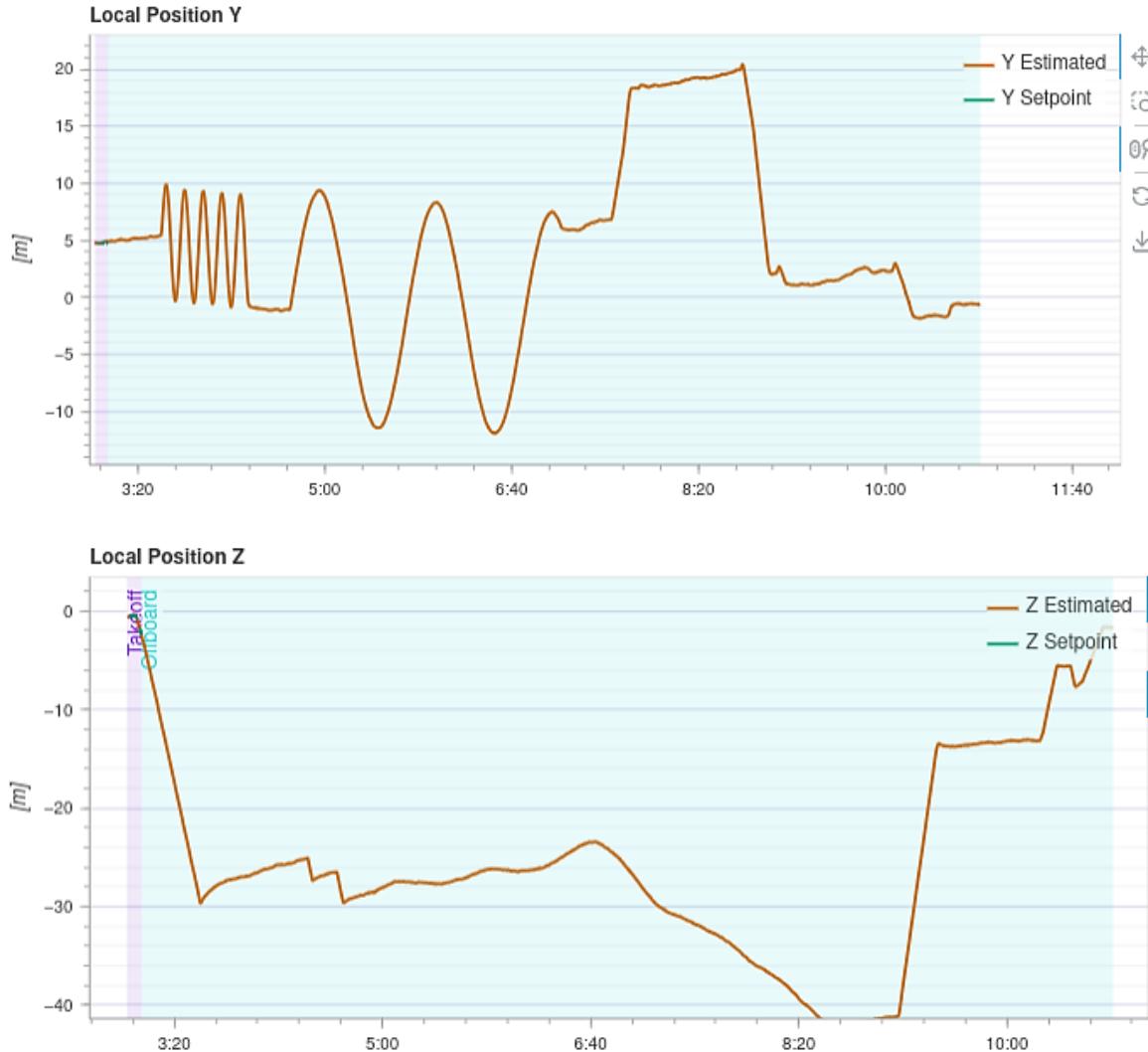
=> Hệ thống an toàn, không kích hoạt chế độ an toàn khẩn cấp (fail-safe) liên quan đến độ cao.

- Đánh giá về Setpoint: Gần như không thay đổi vì điều khiển bằng bàn phím với tổ hợp phím WASD, hệ thống dùng điều khiển theo vận tốc (velocity control) nên không đặt vị trí mục tiêu.

Kết luận: Hệ thống điều khiển Offboard duy trì độ cao chính xác, đáp ứng lệnh bay tốt và không gây mất ổn định

b) Điều khiển vị trí X/Y/Z (Local Position Tracking):





Trong hệ thống PX4 - ROS2, vị trí của drone trong không gian được biểu diễn theo bộ tọa độ cục bộ (X, Y, Z). Mục tiêu của phần đánh giá này là quan sát cách drone di chuyển trong không gian khi được điều khiển bằng bàn phím và phân tích mức độ ổn định, liên tục và an toàn của quá trình di chuyển.

Do phương pháp điều khiển là điều khiển bằng vận tốc (velocity control), hệ thống không đặt Position Setpoint theo dạng waypoint. Vì vậy, các đường Setpoint trong biểu đồ gần như cố định và chỉ đóng vai trò tham chiếu. Đường Estimated thể hiện hoàn toàn chuyển động thực tế của drone.

Những thông số/metrics đánh giá là:

- Local Position Estimated (X/Y/Z) là vị trí thực tế của drone theo từng trục, dùng để đánh giá hướng di chuyển và tính ổn định của quỹ đạo.
- Position Setpoint là vị trí mục tiêu theo từng trục.
- Biên độ thay đổi vị trí cho biết mức độ phản ứng của drone với lệnh điều khiển.

- Độ liên tục của quỹ đạo dùng để đánh giá drone có bị nhảy tọa độ, lệch đột ngột hoặc mất vị trí hay không.

Đánh giá kết quả từ biểu đồ:

- Theo trục X : Vị trí X tăng mạn rồi giảm dần phản ánh đúng thao tác điều khiển, không xuất hiện nhảy tọa độ đột ngột

=> Hệ thống phản hồi chính xác và mượt theo trục X.

- Theo trục Y: Vị trí Y dao động dạng sóng, tròn, trơn tru mượt và không bị nhảy tọa độ đột ngột phản ánh đúng thao tác điều khiển.

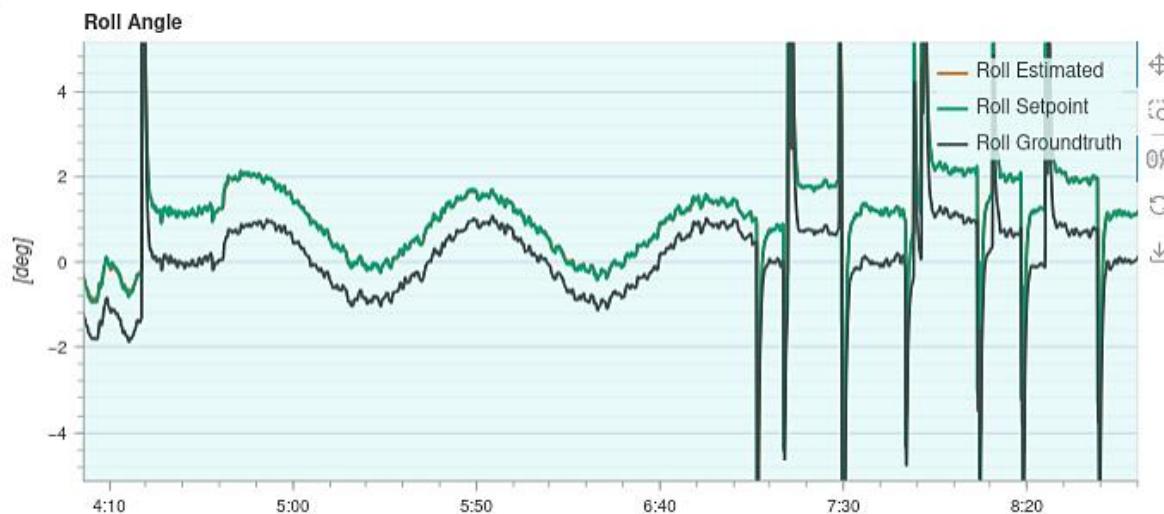
=> Hệ thống phản hồi chính xác và mượt theo trục Y.

- Theo trục Z : Z giảm đến ~ -40m rồi tăng lại gần 0 → thay đổi độ cao có kiểm soát. Không có hiện tượng rơi tự do hoặc tăng độ cao bất thường.

=> Hệ thống phản hồi chính xác và mượt theo trục Z.

Kết luận: Hệ thống điều khiển Offboard duy trì quỹ đạo di chuyển ổn định trong cả ba trục X/Y/Z. Đường Estimated phản ánh đúng lệnh điều khiển từ người vận hành, không xuất hiện dao động bất thường hay mất định vị. Điều này cho thấy hệ thống định vị PX4-ROS2 và bộ lọc hoạt động tin cậy, đảm bảo an toàn trong quá trình điều khiển bàn phím (teleop).

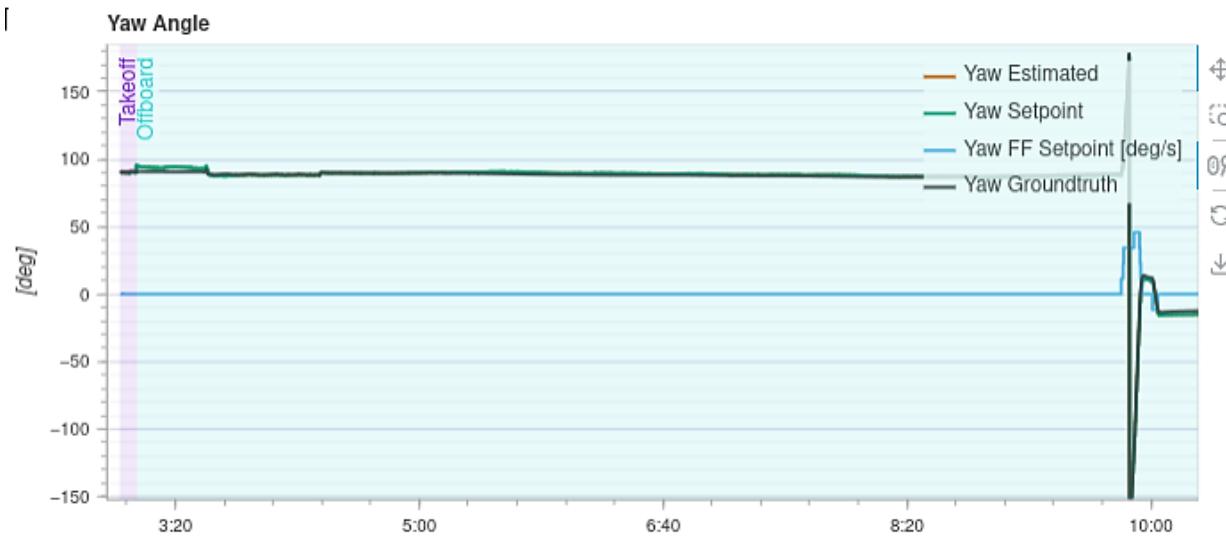
c) Ôn định tư thế (Attitude Tracking):



Biểu đồ Roll Angle : thể hiện góc xoay của drone quanh trục dọc theo hướng bay (trục X) - nghiêng trái hoặc phải.



Biểu đồ Pitch Angle : thể hiện góc quay quanh trục ngang (trục Y) của drone - chói xuống hoặc ngẩng lên theo hướng trước sau.



Biểu đồ Yaw Angle : thể hiện hướng quay theo trục thẳng đứng (trục Z).

Mỗi biểu đồ gồm :

- Estimated – góc thực tế của drone.
- Setpoint – góc mục tiêu hệ thống cố gắng duy trì.
- Groundtruth (nếu có) – giá trị tham chiếu chuẩn

Những thông số/metrics đánh giá là:

- Roll Angle đánh giá ổn định trái phải.
- Pitch Angle đánh giá độ ổn định trước sau.
- Yaw Angle đánh giá độ ổn định khi xoay hướng bay, tránh xoay ngoài ý muốn.

- Biên độ dao động để phát hiện rung hoặc mất cân bằng.

Đánh giá kết quả từ biểu đồ:

- Biểu đồ Roll Angle : Estimated và Setpoint bám sát nhau, sai số nhỏ . Không xuất hiện chênh lệch lớn.

=> Drone giữ ổn định trái phải tốt.

- Biểu đồ Pitch Angle : Có dao động lớn ban đầu do sử dụng chế độ bay SINE sau đó vẫn giữ ổn định khi chuyển sang chế độ bay bằng điều khiển WASD, bám sát Setpoint. Không có chói/ngẩng bất thường.

=> Hệ thống phản hồi nhanh và ổn định.

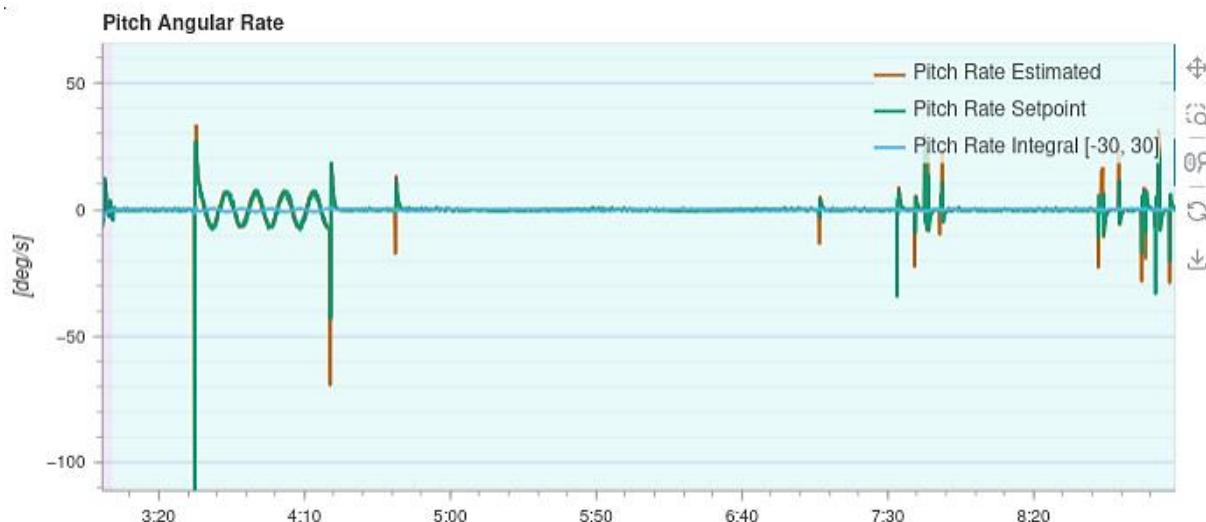
- Biểu đồ Yaw Angle : Góc Yaw gần như giữ nguyên vì không có lệnh xoay kèm theo sai số gần như bằng 0 trong quá trình bay.

=> Không có hiện tượng xoay ngoài ý muốn và drone giữ hướng bay ổn định, không bị xoay lệch.

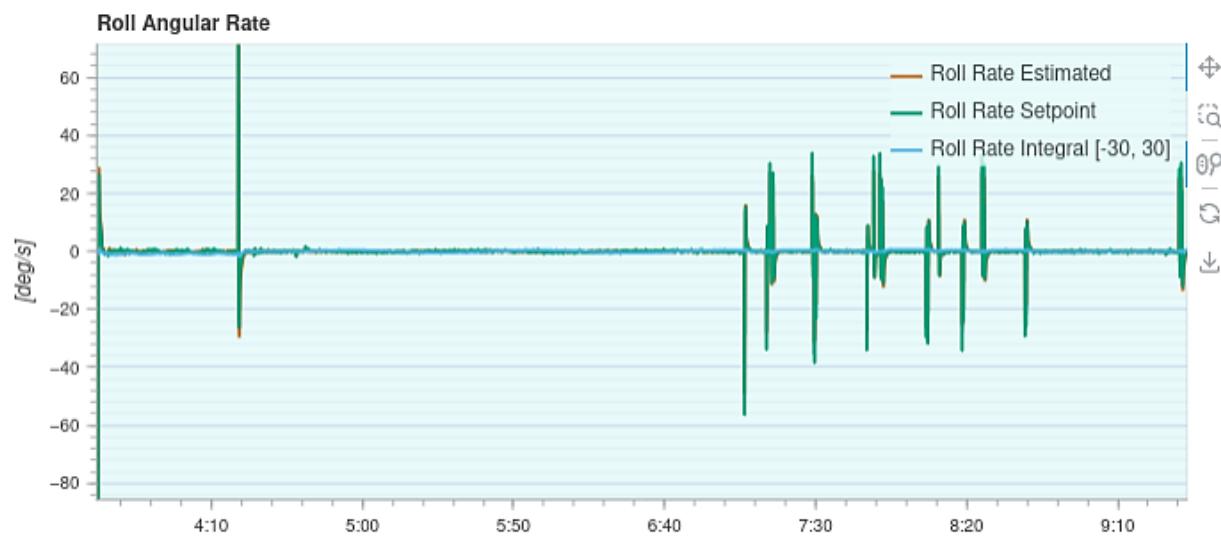
Kết luận: Drone duy trì tư thế ổn định, không gặp mất cân bằng, phù hợp với yêu cầu điều khiển bay mượt.

3.1.2. Độ ổn định bay (Flight Stability):

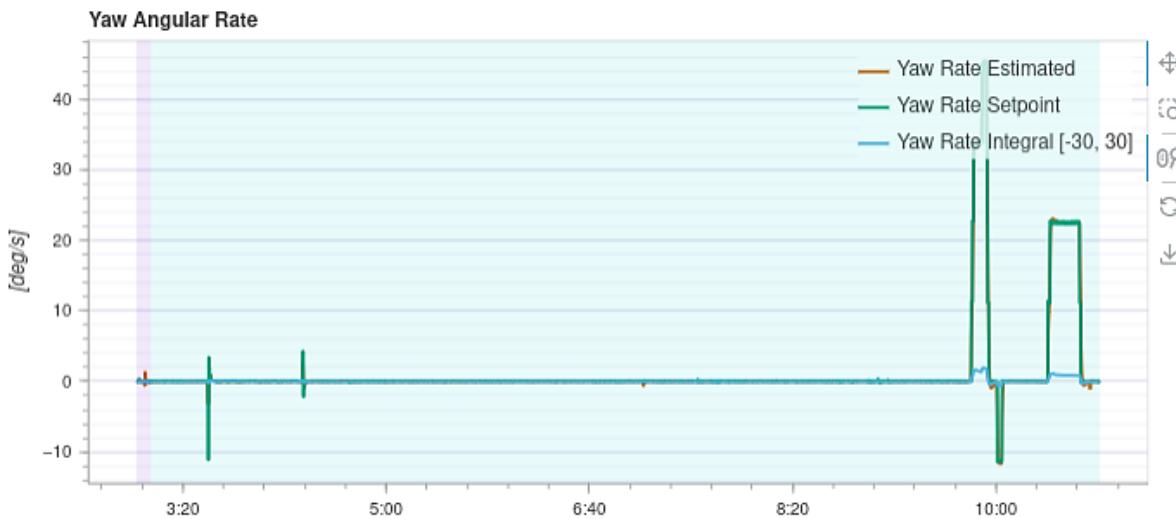
a) Vận tốc góc (Angular Rate Stability) :



Biểu đồ Pitch Angular Rate: biểu thị tốc độ xoay theo hướng Y.



Biểu đồ Roll Angular Rate: biểu thị tốc độ xoay theo hướng X.



Biểu đồ Yaw Angular Rate : biểu thị tốc độ xoay theo hướng Z.

Những thông số / metrics đánh giá :

- Roll Rate : đánh giá mức ổn định trái/phải , phát hiện rung.
- Pitch Rate : đánh giá dao động tiến lùi, kiểm tra khả năng phản hồi.
- Yaw Rate : đánh giá sự ổn định hướng bay, tránh xoay ngoài ý muốn.
- Sai số Estimated – Setpoint: đo mức độ chính xác điều khiển.
- Biên độ dao động : phát hiện rung lắc hoặc mất cân bằng.

Đánh giá kết quả từ biểu đồ:

- Biểu đồ Roll Angular Rate: Xuất hiện vài xung lớn ban đầu khi drone chuyển trạng thái và sau giai đoạn đầu, vận tốc Roll giảm nhanh về gần 0 deg/s. Estimated bám rất sát Setpoint.

=> Drone giữ ổn định trái/phải tốt, không bị rung kéo dài.

- Biểu đồ Pitch Angular Rate: Có dao động lớn trong giai đoạn sớm do quá trình chuyển chế độ bay, sau đó tín hiệu nhanh chóng giảm về 0 deg/s.

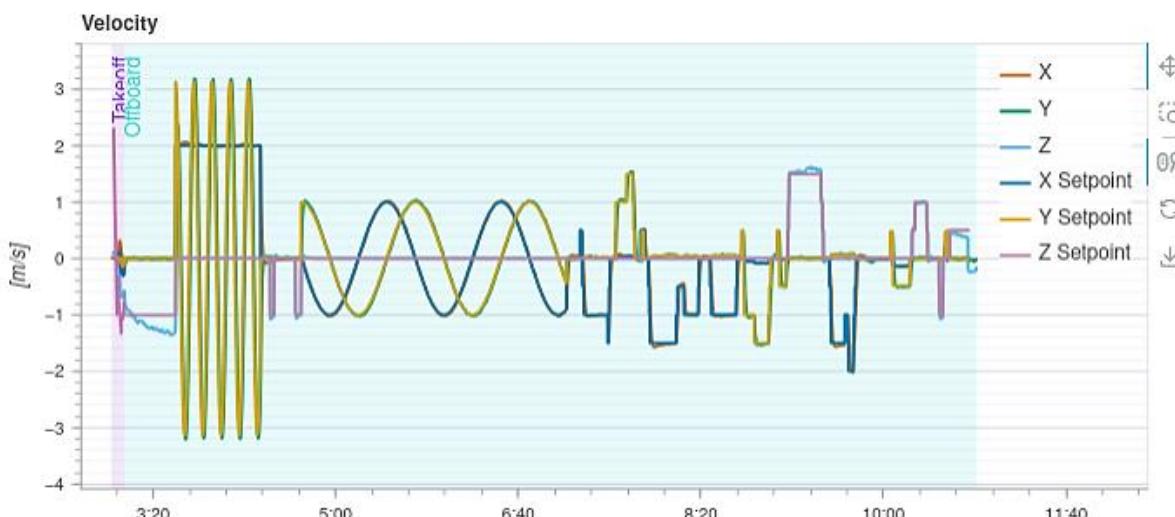
=> Phản hồi nhanh, không xuất hiện chói/ngang mắt kiểm soát.

- Biểu đồ Yaw Angular Rate: Gần như duy trì ở mức 0 deg/s trong toàn bộ chuyến bay, chỉ xuất hiện xung nhỏ khi thay đổi lệnh.

=> Drone không xoay ngoài ý muốn, hướng bay được giữ ổn định.

Kết luận: Drone ổn định rất nhanh sau các dao động ban đầu và vận tốc góc đều duy trì ổn định chứng tỏ không có rung lắc kéo dài và không có sự mất cân bằng. Đây là điểm đặc biệt quan trọng cho việc bay đường dài, giữ camera ổn định và điều khiển chính xác.

b) Vận tốc tuyến tính (Velocity Stability):



Biểu đồ velocity gồm các Setpoint vận tốc mục tiêu được gửi từ ROS2 thông qua lệnh điều khiển và vận tốc thực tế được PX4 ghi lại.

Những thông số metrics đánh giá :

- . X Velocity: vận tốc thực theo hướng trục X.
- . Y Velocity: vận tốc thực theo hướng trục Y.
- . Z Velocity: vận tốc thực theo hướng trục Z.
- . Velocity Setpoint: Mục tiêu để so sánh độ chính xác điều khiển.

Đánh giá kết quả từ biểu đồ:

- Vận tốc trục X: Đường X Velocity và X Setpoint bám sát nhau, dao động rất nhỏ, không xuất hiện chênh hướng ngoài mong muốn.

=> Hệ thống ổn định, hoạt động đúng theo thao tác bay.

- Vận tốc trục Y: Vận tốc thực tế thay đổi theo đúng lệnh điều khiển.

=> Hệ thống giữ cân bằng vận tốc tốt.

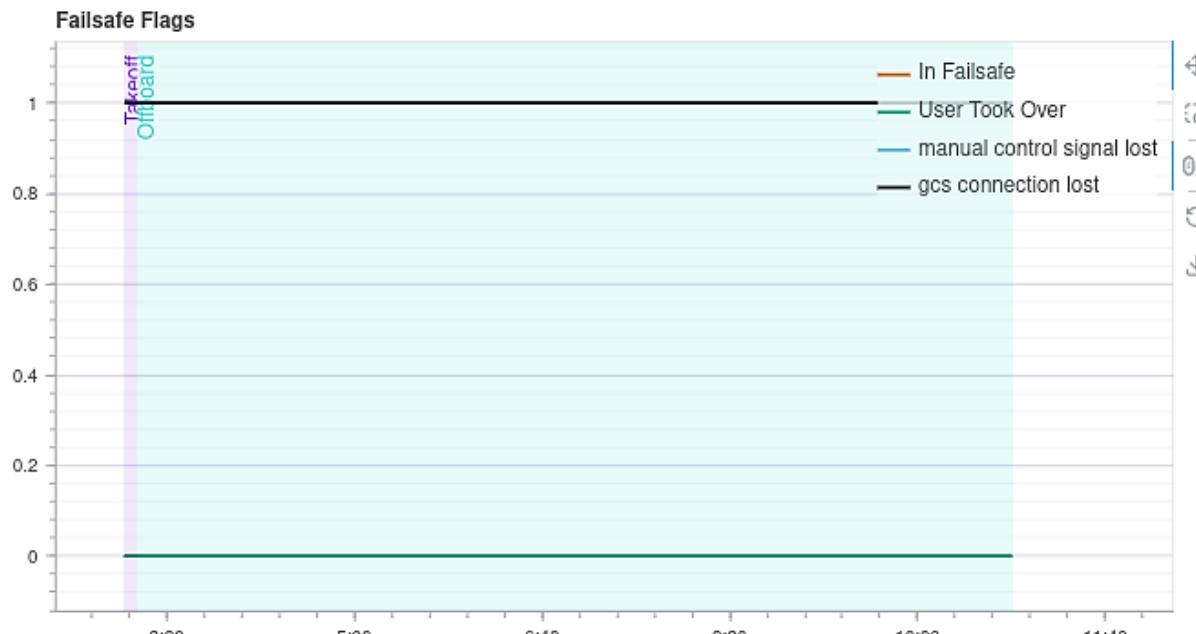
- Vận tốc trục Z: chỉ thay đổi khi có lệnh điều khiển lên xuống, không có tăng tốc đột ngột hoặc rơi tự do.

=> Chứng tỏ hệ thống an toàn và không kích hoạt fail-safe về độ cao

Kết luận: Drone duy trì vận tốc ổn định trên cả 3 trục và các dao động hoạt động đúng theo những thao tác điều khiển.

3.1.3. Mức độ an toàn hệ thống (System Safety):

a) Failsafe Event (Sự kiện an toàn khẩn cấp):



Biểu đồ Failsafe Flags thể hiện các cờ liên quan đến trạng thái an toàn của hệ thống trong suốt quá trình bay gồm 4 tín hiệu :

- In Failsafe: Drone đang trong trạng thái an toàn khẩn cấp.
- User Took Over: Người điều khiển giành quyền điều khiển thủ công.
- Manual Control Signal Lost: Mất tín hiệu điều khiển thủ công.
- GCS connection Lost: Mất kết nối với trạm điều khiển mặt đất.

- Trục thời gian biểu thị toàn bộ thời gian hoạt động từ lúc bắt đầu đến khi hạ cánh.

Các thông số/metrics đánh giá:

Đánh giá kết quả từ biểu đồ:

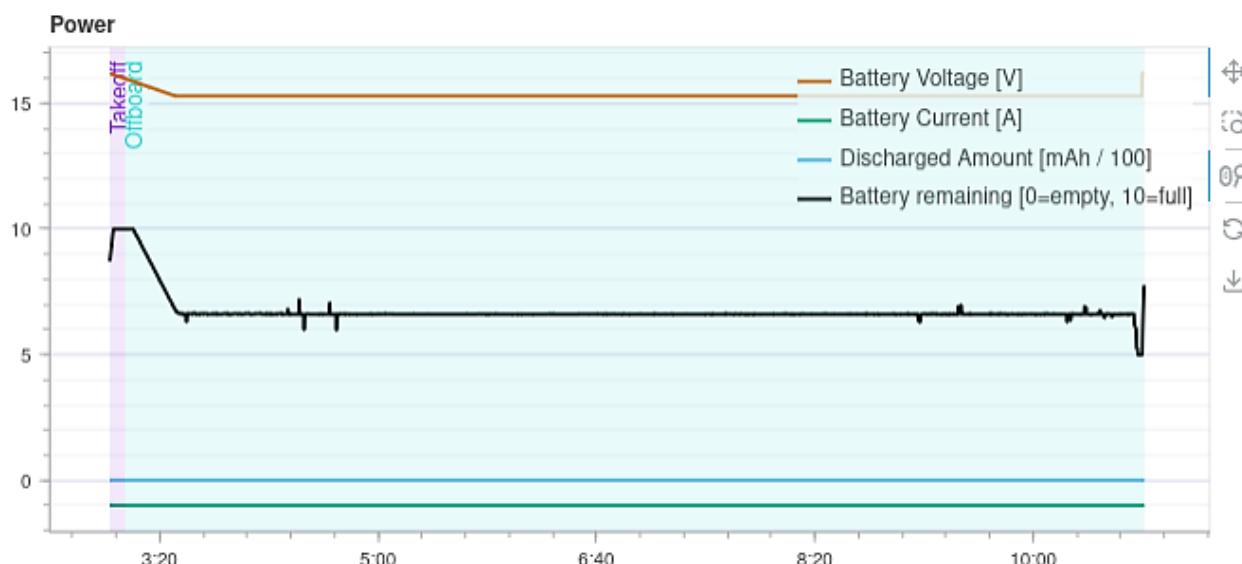
- Gcs connection lost nằm trên ở mức 1 thì Gcs connection là tín hiệu mất kết nối với Qround Control Station nhưng trong chế độ SITL(Software-In-The-Loop là chế độ bay mô phỏng hoàn toàn bằng phần mềm) mô phỏng, PX4 luôn mặc định đánh dấu giá trị này bằng 1 vì SITL không kết nối với phần mềm GCS thực túc là QGroundControl và PX4 coi như không có GCS điều khiển từ xa.

- User Took Over là cờ cho thấy rằng người dùng giành quyền điều khiển qua tay cầm điều khiển. PX4 ghi User Took Over = 0 vì các lệnh được xử lý từ ROS2 (nhận qua phím WASD) được xem là lệnh tự động trong chế độ điều khiển Offboard, không phải điều khiển thủ công. Điều này cho thấy PX4 vẫn giữ toàn bộ quyền điều khiển và giao thức ROS2–PX4 hoạt động ổn định, liên tục. Điều này tốt khi hệ thống được hoạt động xuyên suốt mà không bị gián đoạn.

- Không xuất hiện bất kỳ xung hoặc thay đổi trạng thái trên các đường tín hiệu cho thấy không có sự kiện mất kết nối nào và không có can thiệp khẩn cấp.

Kết luận: Hệ thống điều khiển an toàn và đáng tin cậy trong suốt quá trình bay, việc không kích hoạt bất kỳ Fail-safe nào chứng tỏ cảm biến hoạt động tốt, kết nối ROS2 <-> PX4 <-> MAVLink duy trì tốt, điều khiển không bị gián đoạn và đủ điều kiện cho các bài bay dài hơn và phức tạp hơn.

b) Pin và rung động (Battery & Vibration):



Biểu đồ thể hiện các nguồn điện của drone theo thời gian :

- Battery Voltage (V): thể hiện điện áp nguồn cấp cho hệ thống, dùng để theo dõi sụt áp và phát hiện nguy cơ tắt nguồn.
- Battery Current (A): thể hiện dòng điện tiêu thụ trong quá trình bay, dùng để đánh giá tải hệ thống và mức tiêu thụ điện.
- Discharged Amount: thể hiện lượng điện đã tiêu hao, dùng để theo dõi tốc độ tiêu thụ pin.
- Battery Remaining: Mức pin còn lại, cho thấy khả năng duy trì thời gian bay.

Đánh giá kết quả từ đồ thị:

- Điện áp pin giảm nhẹ theo thời gian nhưng theo xu hướng tuyến tính, không xuất hiện hiện tượng “tụt áp” đột ngột.

=> chứng tỏ hệ thống không bị quá tải và nguồn cấp luôn ở trạng thái an toàn.

- Giá trị Battery Current duy trì xấp xỉ mức 0A. Đây là hành vi đặc trưng của mô phỏng SITL, do không có động cơ vật lý tiêu thụ điện.

=> không gây nguy cơ quá tải bộ điều khiển.

- Thông số Discharged Amount gần như không thay đổi vì SITL không mô phỏng quá trình xả pin thực tế.

=> Điều này khẳng định hệ thống vẫn hoạt động bình thường và không gặp rủi ro cạn pin.

- Trong toàn bộ chuyến bay, hệ thống không rơi vào trạng thái failsafe liên quan đến nguồn điện.

=> Drone có thể tiếp tục hoạt động và ổn định

Kết luận: Hệ thống điều khiển Offboard vận hành với nguồn điện ổn định, không có dấu hiệu sụt áp, quá tải hoặc tiêu hao bất thường. Dữ liệu gần bằng 0 của dòng tiêu thụ và lượng xả pin là đặc trưng của môi trường mô phỏng SITL, không phải lỗi hệ thống. Điều này cho thấy drone hoạt động trong điều kiện tốt và không phát sinh nguy cơ mất nguồn trong quá trình bay.

3.2. Mô tả các files kết quả lưu trữ thông tin:

Trong quá trình triển khai hệ thống điều khiển phương tiện bay không người lái (drone) bằng nền tảng Robot Operating System phiên bản 2 (ROS2), kết hợp với mô phỏng bằng phần mềm điều khiển bay PX4 (PX4 Software-In-The-Loop Simulation – PX4 SITL) và môi trường mô phỏng Gazebo, hệ thống tạo ra nhiều loại tệp kết quả và tệp nhật ký khác nhau. Các tệp này lưu trữ thông tin quan trọng phục vụ cho việc giám sát, đánh giá hiệu năng, phân tích lỗi và xác minh hoạt động của hệ thống. Phần này trình bày chi tiết chức năng của từng loại tệp, nội dung được lưu trữ bên trong cũng như vai trò của chúng trong quá trình đánh giá hệ thống.

3.2.1. File log chính của PX4:

Vị trí lưu file: PX4-Autopilot/build/px4_sitl_default/rootfs/log/<ngày>/

Chức năng: Lưu toàn bộ dữ liệu chuyến bay (Flight Data; ghi lại cảm biến, trạng thái điều khiển, thông số hệ thống; là nguồn dữ liệu đầu vào cho Flight Review.

Kết luận: Đây là file quan trọng nhất để đánh giá hệ thống. Mỗi lần chạy mô phỏng tạo ra một file .ulg tương ứng với một chuyến bay.

3.2.2. Logged Messages (Nhật ký sự kiện):

Logged Messages

#	Time	Level	Message
0	0:02:57	INFO	Armed by external command
1	0:02:57	INFO	Using default takeoff altitude: 2.50 m
2	0:02:57	INFO	logging: opening log file 2025-11-24/6_53_24.ulg
3	0:02:57	INFO	[logger] Start file log (type: full)
4	0:02:57	INFO	[logger] closed logfile, bytes written: 2455083
5	0:02:57	INFO	[logger] Opened full log file: ./log/2025-11-24/06_53_24.ulg
6	0:02:58	INFO	Takeoff detected
7	0:03:02	WARNING	[timesync] time jump detected. Resetting time synchroniser.
8	0:03:02	WARNING	[uxrce_dds_client] time sync no longer converged
9	0:03:04	INFO	[uxrce_dds_client] time sync converged
10	0:03:29	WARNING	[timesync] time jump detected. Resetting time synchroniser.

Logged Messages là bảng ghi lại toàn bộ sự kiện hệ thống trong quá trình bay, bao gồm:

- Thời điểm xảy ra sự kiện
- Mức độ thông báo: INFO / WARNING / ERROR
- Nội dung sự kiện: ARM, TAKEOFF, log start, time sync, cảnh báo,...

Vai trò của Logged Messages:

- Giúp theo dõi tiến trình bay :

+ Giúp xác nhận hệ thống đi đúng chu trình trạng thái bay FSM (Finite State Machine) hay không: IDLE → ARMING → TAKEOFF → OFFBOARD → LAND

Trong đó:

- IDLE là trạng thái nghỉ, drone chưa kích hoạt động cơ chỉ nằm yên và sẵn sàng nhận lệnh.
- ARMING là trạng thái kích hoạt động cơ, drone đang bật hệ thống, kiểm tra an toàn và chuyển sang trạng thái có thể cất cánh, ngược lại là DISARM là tắt động cơ.
- TAKEOFF là trạng thái cất cánh, drone bắt đầu rời mặt đất và tăng độ cao.

- OFFBOARD là chế độ điều khiển từ bên ngoài, drone không tự bay mà nhận lệnh từ máy tính/ROS2.

- LAND là trạng thái hạ cánh.

+ Kiểm tra xem lệnh ARM và TAKEOFF có được thực thi đúng thời điểm hay không.

- Phát hiện bất thường: Các dòng WARNING phản ánh vấn đề tiềm ẩn như

+ Lỗi thời gian

+ Mất đồng bộ DDS(Data Distribution Service là giao thức để nhận và gửi dữ liệu)

+ Trục trặc đồng bộ hệ thống

Nếu có ERROR có thể dẫn tới Fail-safe.

- Đánh giá an toàn: Có thể dựa vào để đánh giá hệ thống duy trì ổn định.

- Hỗ trợ phân tích hậu kỳ: Logged Messages giúp truy vết nguyên nhân sự cố khi bay sai hoặc mất tính ổn định hoặc sai lệnh.

3.2.3. Non-default Parameters (Các tham số cấu hình thay đổi):

Non-default Parameters (except RC and sensor calibration)

#	Name	Value	Frame Default	Min	Max	Description
3	BAT_V_CHARGED	4.050000190734863	4.05000019073			(unknown)
4	BAT_V_EMPTY	3.599999046325684	3.5999990463			(unknown)
5	BAT_V_LOAD_DROP	0.30000001192092896	0.30000001192			(unknown)
6	CA_ROTOR0_PX	0.13	0.13	-100	100	Position of rotor 0 along X body axis relative to center of gra
7	CA_ROTOR0_PY	0.22	0.22	-100	100	Position of rotor 0 along Y body axis relative to center of gra
8	CA_ROTOR1_PX	-0.13	-0.13	-100	100	Position of rotor 1 along X body axis relative to center of gra
9	CA_ROTOR1_PY	-0.2	-0.2	-100	100	Position of rotor 1 along Y body axis relative to center of gra
10	CA_ROTOR2_KM	-0.05	-0.05	-1	1	Moment coefficient of rotor 2
11	CA_ROTOR2_PX	0.13	0.13	-100	100	Position of rotor 2 along X body axis relative to center of gra
12	CA_ROTOR2_PY	-0.22	-0.22	-100	100	Position of rotor 2 along Y body axis relative to center of gra
13	CA_ROTOR3_KM	-0.05	-0.05	-1	1	Moment coefficient of rotor 3

Bảng Non-default Parameters ghi lại những tham số trong PX4 đã bị thay đổi so với giá trị mặc định của hệ thống. Các tham số này có thể liên quan đến: Pin & nguồn điện (BAT_V_CHARGED, BAT_V_EMPTY,...), cấu hình phần cơ khí (vị trí rotor dọc theo trục X/Y), hệ số mô-men của rotor (rotor moment coefficient), các tham số hiệu chuẩn cảm biến. Có vai trò :

+ Xác nhận drone được thiết lập đúng tiêu chuẩn

+ Không có thay đổi bất thường có thể gây mất ổn định.

+ Đảm bảo bài thực nghiệm có thể tái lập và kiểm chứng.

3.2.4. Performance Counters (Thông kê hiệu năng hệ thống):

Performance Counters

Pre Flight:

```
mission_dm_cache_miss: 0 events
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
logger_sd_fsync_mission: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.0
logger_sd_write_mission: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.0
logger_sd_fsync: 27 events, 8000us elapsed, 296.30us avg, min 0us max 4000us 32
logger_sd_write: 462 events, 0us elapsed, 0.00us avg, min 0us max 0us 492.735us
mavlink: STATUSTEXT missed messages: 0 events
mavlink: forwarding error: 0 events
mavlink: send_bytes error: 0 events
mavlink: tx run interval: 2509 events, 3998.41 avg, min 4000us max 4000us 0.000
mavlink: tx run elapsed: 2509 events, 0us elapsed, 0.00us avg, min 0us max 0us
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
mavlink: STATUSTEXT missed messages: 0 events
mavlink: forwarding error: 0 events
mavlink: send_bytes error: 0 events
mavlink: tx run interval: 2509 events, 3998.41 avg, min 4000us max 4000us 0.000
mavlink: tx run elapsed: 2509 events, 0us elapsed, 0.00us avg, min 0us max 0us
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
mavlink: STATUSTEXT missed messages: 0 events
mavlink: forwarding error: 0 events
mavlink: send_bytes error: 0 events
mavlink: tx run interval: 2509 events, 3998.41 avg, min 4000us max 4000us 0.000
mavlink: tx run elapsed: 2509 events, 0us elapsed, 0.00us avg, min 0us max 0us
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
mavlink: STATUSTEXT missed messages: 0 events
mavlink: forwarding error: 0 events
mavlink: send_bytes error: 0 events
mavlink: tx run interval: 2509 events, 3998.41 avg, min 4000us max 4000us 0.000
mavlink: tx run elapsed: 2509 events, 0us elapsed, 0.00us avg, min 0us max 0us
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
gyro_calibration: calibration updated: 0 events
gyro_calibration: interval: 3 events, 3333333.33 avg, min 8000us max 9992000us
gyro_fft: gyro data gap: 0 events
gyro_fft: FFT: 60 events, 0us elapsed, 0.00us avg, min 0us max 0us 520.455us rm
gyro_fft: cycle interval: 627 events, 15974.48 avg, min 16000us max 16000us 0.0
gyro_fft: cycle: 627 events, 0us elapsed, 0.00us avg, min 0us max 0us 276.837us
uxrce_dds_client: cycle interval: 3794 events, 2643.12 avg, min 0us max 8000us
uxrce_dds_client: cycle: 3793 events, 10028000us elapsed, 2643.82us avg, min 0u
rtl_dm_cache_miss_land: 0 events
DatamanClient: sync: 0 events, 0us elapsed, 0.00us avg, min 0us max 0us 0.000us
rtl_dm_cache_miss_geo: 0 events
```

Performance Counters là nhóm dữ liệu ghi lại số liệu hoạt động của các module bên trong PX4 trong suốt quá trình bay. Các thông tin này giúp đánh giá hiệu suất xử lý của hệ thống

điều khiển, kiểm tra mức độ ổn định của cảm biến và giao tiếp, phát hiện lỗi quá tải hoặc treo (stall) trong các tiến trình, xác nhận hệ thống hoạt động bình thường và không bị gián đoạn.

Performance Counters bao gồm các thông kê chính:

- MAVLink Errors: Lỗi truyền nhận dữ liệu giữa PX4 và GCS.
- Sensor Processing Cycles: Tần suất xử lý dữ liệu cảm biến.
- Controller Cycle Time : Chu kỳ xử lý của bộ điều khiển.
- Data Sync & Logging: Việc ghi log và đồng bộ dữ liệu

Dữ liệu được ghi hai giai đoạn chính :

- Pre-Flight : trước khi cất cánh.
- Post-Flight: sau khi drone hoàn thành nhiệm vụ.

Ý nghĩa đối với an toàn hệ thống: Performance Counters cho thấy hệ thống không gặp quá tải trong quá trình bay, không có module nào bị treo hoặc mất phản hồi, cảm biến và bộ điều khiển hoạt động liên tục và chính xác, không xuất hiện tình trạng lỗi kéo dài dẫn đến failsafe.

Kết luận: Hệ thống Offboard vận hành ổn định, đáng tin cậy và đáp ứng đầy đủ yêu cầu an toàn trong thí nghiệm bay.

3.2.5. Flight Review Charts (Biểu đồ phân tích dữ liệu):

Được sinh ra từ file .ulg

Bao gồm: Độ cao (Altitude), vị trí X/Y/Z, tư thế Roll/Pitch/Yaw, vận tốc, pin và Fail-safe

Có vai trò: Hiểu rõ hành vi của drone trong suốt quá trình bay, phát hiện sự cố tiềm ẩn (rung, lệch hướng, mất độ cao, quá tải); đánh giá chất lượng điều khiển Offboard; kiểm chứng tính ổn định và chính xác của thuật toán/hệ thống

Đặc biệt, việc phân tích biểu đồ cho phép so sánh Estimated vs Setpoint để xác định sai số điều khiển, kiểm tra sự nhất quán giữa các cảm biến và theo dõi trạng thái bay theo từng giai đoạn.

Kết luận: Các biểu đồ từ Flight Review là thành phần quan trọng trong phân tích dữ liệu chuyến bay. Chúng cung cấp cái nhìn tổng thể về độ ổn định bay, hiệu năng điều khiển, an toàn hệ thống và tính chính xác của cảm biến và thuật toán.

3.3. Kết quả demo:

a) Các bước chạy chương trình:

Bước 1: Nạp môi trường ROS2 cho gói phần mềm đã xây dựng bằng câu lệnh:

```
source install/setup.bash
```

Giúp cho ROS2 nhận diện được các package px4_offboard và nhận diện các khói gồm: velocity_control, visualizer, control, processes.

Bước 2: Chạy file offboard_velocity_control.launch.py (nằm trong thư mục lauch/) bằng câu lệnh:

```
ros2 launch px4_offboard offboard_velocity_control.launch.py
```

Từ đó sẽ tự động chạy các khói:

- Khối visualizer: Hiển thị quỹ đạo của drone, vị trí, vận tốc tên RViz2, giúp theo dõi dữ liệu bay theo thời gian thật.
- Khối processes: Tự động mở terminal để chạy Micro XRECE-DDS Agent (là thành phần trung gian bắt buộc để ROS2 giao tiếp với PX4 khi chạy ở chế độ điều khiển Offboard theo tiêu chuẩn DDS) và PX4 SITL (mô phỏng drone - Gazebo X500).
- Khối control: Đọc bàn phím điều khiển, gửi lệnh vận tốc và tín hiệu arm/disarm và cho phép bay theo chế độ SINE/CIRCLE/MANUAL.
- Khối velocity_control: là trung tâm điều khiển Offboard quản lý các trạng thái: IDLE → ARMING → TAKEOFF → LOITER → OFFBOARD, gửi TrajectorySetpoint để PX4 bay theo (TrajectorySetpoint là một gói tin chứa một tập thông số mô tả quỹ đạo mong muốn bao gồm: vận tốc, hướng (yaw) mà ta muốn) đồng thời hiển thị NAV_STATUS, ARM_STATUS, FAILSAFE
- Hiển thị Gazebo: mở giao diện trực quan để xem drone và đường bay.

Kết quả sau khi chạy:

- Đầu tiên là terminal chính (bộ não của PX4).

```

INFO [commander] LED: open /dev/led0 failed (22)
WARN [health_and_arming_checks] Preflight Fail: ekf2 missing data
INFO [tone_alarm] home set
INFO [uxrce_dds_client] init UDP agent IP:127.0.0.1, port:8888
INFO [mavlink] mode: Normal, data rate: 4000000 B/s on udp port 18570 remote port 14550
INFO [mavlink] mode: Onboard, data rate: 4000000 B/s on udp port 14580 remote port 14540
INFO [mavlink] mode: Onboard, data rate: 4000 B/s on udp port 14280 remote port 14030
INFO [mavlink] mode: Gimbal, data rate: 400000 B/s on udp port 13030 remote port 13280
INFO [logger] logger started (mode=all)
INFO [logger] Start file log (type: full)
INFO [logger] [logger] ./log/2025-11-19/14_31_57.ulg
INFO [logger] Opened full log file: ./log/2025-11-19/14_31_57.ulg
INFO [mavlink] MAVLink only on localhost (set param MAV_{i}_BROADCAST = 1 to enable network)
INFO [mavlink] MAVLink only on localhost (set param MAV_{i}_BROADCAST = 1 to enable network)
INFO [px4] Startup script returned successfully
pxh> INFO [tone_alarm] notify positive
INFO [commander] Ready for takeoff!
INFO [uxrce_dds_client] synchronized with time offset 1763559842064507us

```

=> Cho ta thấy toàn bộ dữ liệu chuyến bay

```

INFO [logger] logger started (mode=all)
INFO [logger] Start file log (type: full)
INFO [logger] [logger] ./log/2025-11-19/14_31_57.ulg
INFO [logger] Opened full log file: ./log/2025-11-19/14_31_57.ulg

```

PX4 tích hợp một thành phần gọi là PX4 Logger, dùng để ghi lại toàn bộ điều khiển và trạng thái bay và thông điệp lưu vào một file có định dạng là ULog

- PX4 gửi dữ liệu MAVLink -> QGroundControl -> Gazebo : PX4 SITL tự động khởi tạo nhiều phiên bản MAVLink, mỗi phiên phục vụ cho một mục đích khác nhau.

```

INFO [uxrce_dds_client] init UDP agent IP:127.0.0.1, port:8888
INFO [mavlink] mode: Normal, data rate: 4000000 B/s on udp port 18570 remote port 14550
INFO [mavlink] mode: Onboard, data rate: 4000000 B/s on udp port 14580 remote port 14540
INFO [mavlink] mode: Onboard, data rate: 4000 B/s on udp port 14280 remote port 14030
INFO [mavlink] mode: Gimbal, data rate: 400000 B/s on udp port 13030 remote port 13280

```

Kết nối chính giữa PX4 và QGroundControl ở port local :18570 và port remote (QGC) :14550 tốc độ 4MB/s.

- Giao tiếp với ROS2 qua XRCE-DDS

```

WARN [health_and_arming_checks] Preflight Fail: ekf2 missing data
INFO [tone_alarm] home set
INFO [uxrce_dds_client] init UDP agent IP:127.0.0.1, port:8888
INFO [mavlink] mode: Normal, data rate: 4000000 B/s on udp port 18570 remote port 14550

```

- In ra các trạng thái :

Trạng thái PX4 đã xác nhận vị trí bay , thỏa điều kiện bay

```
[INFO [commander] Ready for takeoff!
```

- Trạng thái PX4 không ước lượng được hướng của drone , tức là do sai tư thế nên không thể thỏa điều kiện bay.

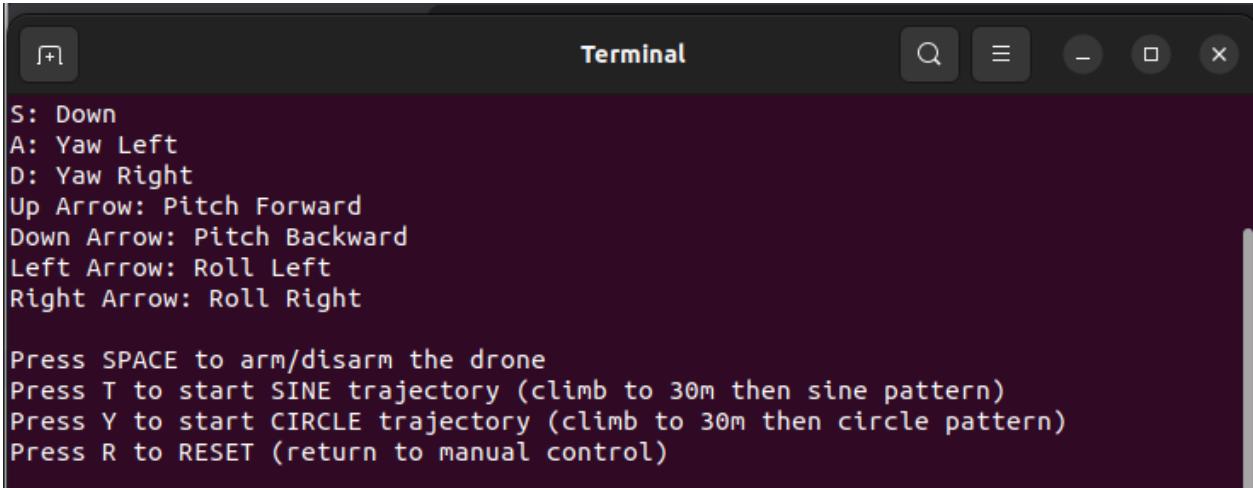
```
[1763564453.965827] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, topic_id: 0x0FC(2), participant_id: 0x001(1)
[1763564453.966012] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, publisher_id: 0x0FC(3), participant_id: 0x001(1)
[1763564453.966121] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, datawriter_id: 0x0FC(5), publisher_id: 0x0FC(3)
[1763564453.966145] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x100(2), participant_id: 0x001(1)
[1763564453.966343] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x100(5), publisher_id: 0x100(3)
[1763564453.966496] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x105(2), participant_id: 0x001(1)
[1763564453.966526] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x105(3), participant_id: 0x001(1)
[1763564453.966861] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x105(5), publisher_id: 0x105(3)
[1763564453.966997] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x10A(2), participant_id: 0x001(1)
[1763564453.967029] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x10A(3), participant_id: 0x001(1)
[1763564453.967261] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x10A(5), publisher_id: 0x10A(3)
```

- Thứ 2 là terminal hiển thị quá trình khởi tạo các thực thể DDS (Data Distribution Service) để gửi và nhận thông điệp như Topics, Publishers, Subscribers, DataReaders và DataWriters . Mỗi thực thể như vậy tương ứng với một kênh dữ liệu mà PX4 sử dụng để gửi trạng thái về vận tốc cũng như hướng.

```
Terminal
[1763564453.965827] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, topic_id: 0x0FC(2), participant_id: 0x001(1)
[1763564453.966012] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, publisher_id: 0x0FC(3), participant_id: 0x001(1)
[1763564453.966121] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, datawriter_id: 0x0FC(5), publisher_id: 0x0FC(3)
[1763564453.966145] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x100(2), participant_id: 0x001(1)
[1763564453.966343] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x100(5), publisher_id: 0x100(3)
[1763564453.966496] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x105(2), participant_id: 0x001(1)
[1763564453.966526] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x105(3), participant_id: 0x001(1)
[1763564453.966861] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x105(5), publisher_id: 0x105(3)
[1763564453.966997] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x10A(2), participant_id: 0x001(1)
[1763564453.967029] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x10A(3), participant_id: 0x001(1)
[1763564453.967261] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x10A(5), publisher_id: 0x10A(3)

Terminal
[1763564453.966861] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, topic_id: 0x0FC(2), participant_id: 0x001(1)
[1763564453.966997] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, publisher_id: 0x0FC(3), participant_id: 0x001(1)
[1763564453.967029] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, datawriter_id: 0x0FC(5), publisher_id: 0x0FC(3)
[1763564453.967261] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x100(2), participant_id: 0x001(1)
[1763564453.967391] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x100(5), publisher_id: 0x100(3)
[1763564453.967521] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x105(2), participant_id: 0x001(1)
[1763564453.967651] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x105(3), participant_id: 0x001(1)
[1763564453.967781] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x105(5), publisher_id: 0x105(3)
[1763564453.967911] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x00000001, topic_id: 0x10A(2), participant_id: 0x001(1)
[1763564453.968041] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x00000001, publisher_id: 0x10A(3), participant_id: 0x001(1)
[1763564453.968171] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x00000001, datawriter_id: 0x10A(5), publisher_id: 0x10A(3)
```

- Thứ 3 là terminal của node control.py, đây là nơi nhận dữ liệu từ bàn phím và xuất lệnh để điều khiển drone, nó là một Teleop node - điều khiển drone bằng keyboard.



```

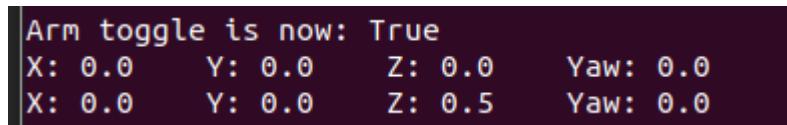
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

```

Nhận input từ bàn phím (W, A, S, D, mũi tên...) → Điều khiển thủ công (Manual Offboard). Nhấn Space để khởi động hoặc tắt. Nhấn T để kích hoạt quỹ đạo SINE. Nhấn Y để kích hoạt quỹ đạo hình tròn. Nhấn R để reset quay về chế độ điều khiển thủ công.

- Khi nó nhận lệnh thủ công thì các thông số tọa độ và vận tốc được hiện lên :

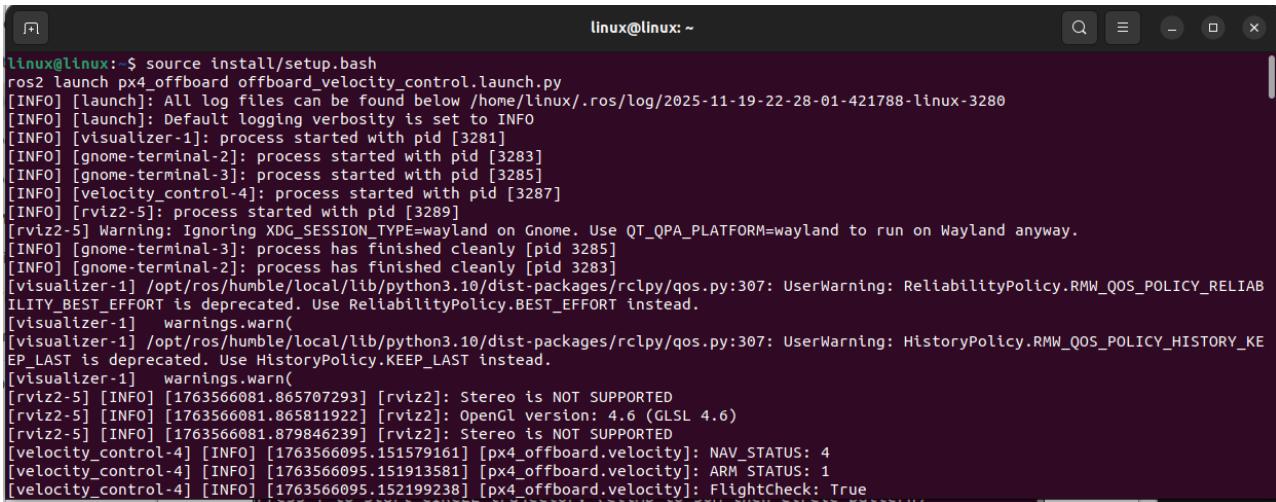


```

Arm toggle is now: True
X: 0.0      Y: 0.0      Z: 0.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.5      Yaw: 0.0

```

- Thứ 4 terminal này là node của velocity_control chịu trách nhiệm nhận lệnh người dùng từ control.py qua các kênh (topic) : /arm_message , /trajectory_mode, /offboard_velocity_cmd và tạo trajectory Setpoints gửi cho PX4 để bay theo điều khiển .



```

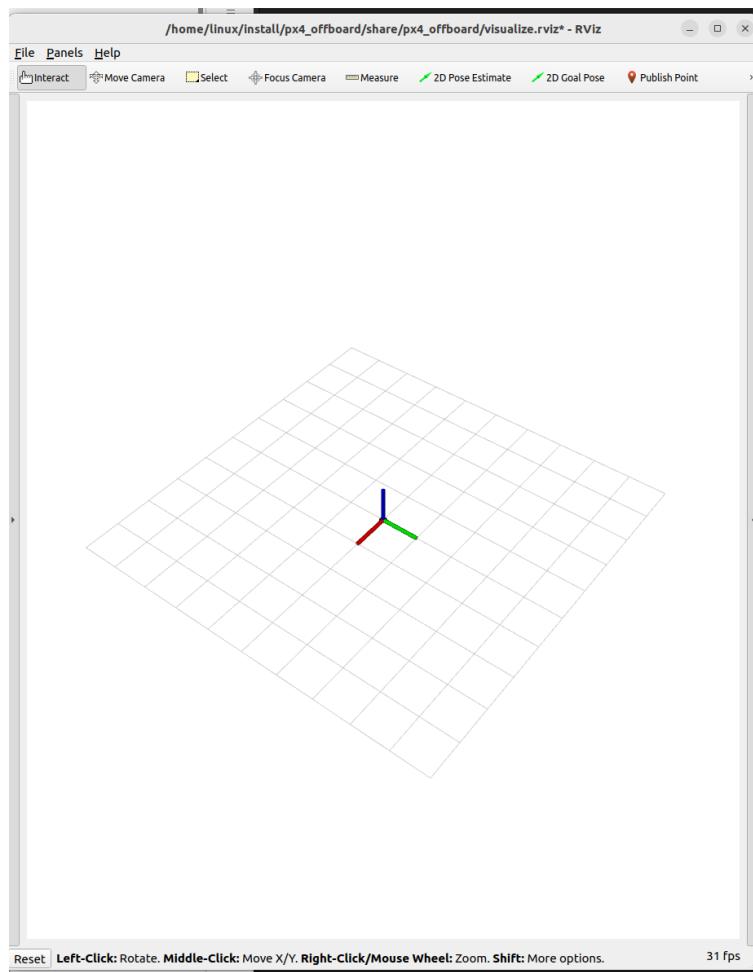
linux@linux: $ source install/setup.bash
ross2 launch px4_offboard offboard_velocity_control.launch.py
[INFO] [launch]: All log files can be found below /home/linux/.ros/log/2025-11-19-22-28-01-421788-linux-3280
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [visualizer-1]: process started with pid [3281]
[INFO] [gnome-terminal-2]: process started with pid [3283]
[INFO] [gnome-terminal-3]: process started with pid [3285]
[INFO] [velocity_control-4]: process started with pid [3287]
[INFO] [rviz2-5]: process started with pid [3289]
[rviz2-5] Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland anyway.
[INFO] [gnome-terminal-3]: process has finished cleanly [pid 3285]
[INFO] [gnome-terminal-2]: process has finished cleanly [pid 3283]
[visualizer-1] /opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/qos.py:307: UserWarning: ReliabilityPolicy.RMW_QOS_POLICY_RELIABILITY_BEST_EFFORT is deprecated. Use ReliabilityPolicy.BEST_EFFORT instead.
[visualizer-1] warnings.warn(
[visualizer-1] /opt/ros/humble/local/lib/python3.10/dist-packages/rclpy/qos.py:307: UserWarning: HistoryPolicy.RMW_QOS_POLICY_HISTORY_KEEP_LAST is deprecated. Use HistoryPolicy.KEEP_LAST instead.
[visualizer-1] warnings.warn(
[rviz2-5] [INFO] [1763566081.865707293] [rviz2]: Stereo is NOT SUPPORTED
[rviz2-5] [INFO] [1763566081.865811922] [rviz2]: OpenGL version: 4.6 (GLSL 4.6)
[rviz2-5] [INFO] [1763566081.879846239] [rviz2]: Stereo is NOT SUPPORTED
[velocity_control-4] [INFO] [1763566095.151579161] [px4_offboard.velocity]: NAV_STATUS: 4
[velocity_control-4] [INFO] [1763566095.151913581] [px4_offboard.velocity]: ARM STATUS: 1
[velocity_control-4] [INFO] [1763566095.152199238] [px4_offboard.velocity]: FlightCheck: True

```

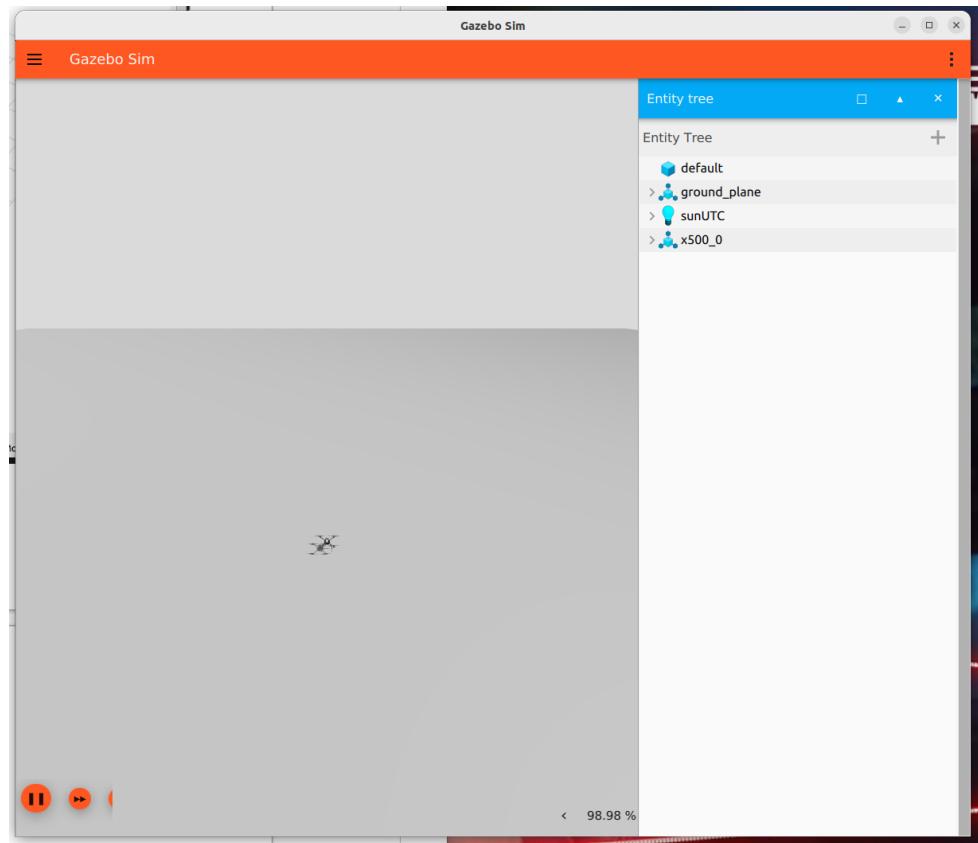
Liên tục kiểm tra và hiển thị trạng thái như “NAV_STATUS: 4” tức là PX4 đang ở trạng

thái ổn định và “ARM STATUS :1” là drone đang disarmed và “FlightCheck : True” là vừa nhấn space tức là gửi yêu cầu ARM, “ARM STATUS :2” là drone đã được khởi động, “NAV_STATUS: 17” là PX4 xác nhận đang trong trạng thái cất cánh (Takeoff).

- Thứ 5 là cửa sổ RViz2 được dùng để trực quan hóa các dữ liệu OFFBOARD như vị trí thực tế của drone, vector vận tốc, setpoint quỹ đạo và TF frames. Khi drone bắt đầu bay Offboard, RViz hiển thị quỹ đạo (Path), vector hướng (Arrow) và khung tọa độ của drone trong không gian 3D. Điều này giúp kiểm tra xem thuật toán điều khiển OFFBOARD có chạy chính xác hay không. Trục màu xanh nước biển biểu thị cho z, trục màu đỏ biểu thị cho y, trục màu xanh lá cây biểu thị cho x.



- Thứ 6, cửa sổ Gazebo Sim là môi trường mô phỏng vật lý dùng để chạy PX4 SITL. Trong Gazebo, drone X500_0 được mô phỏng đầy đủ động lực học và cảm biến, cho phép thử nghiệm thuật toán điều khiển Offboard mà không cần phần cứng thật. Entity Tree hiển thị các đối tượng trong mô phỏng như mặt đất, ánh sáng và mô hình drone. PX4 SITL nhận sensor từ Gazebo và gửi lại lệnh motor, tạo thành vòng lặp bay hoàn chỉnh.



b) Quá trình demo chương trình:

- Thực hiện thao tác bay lên hoặc bay xuống theo trục z:

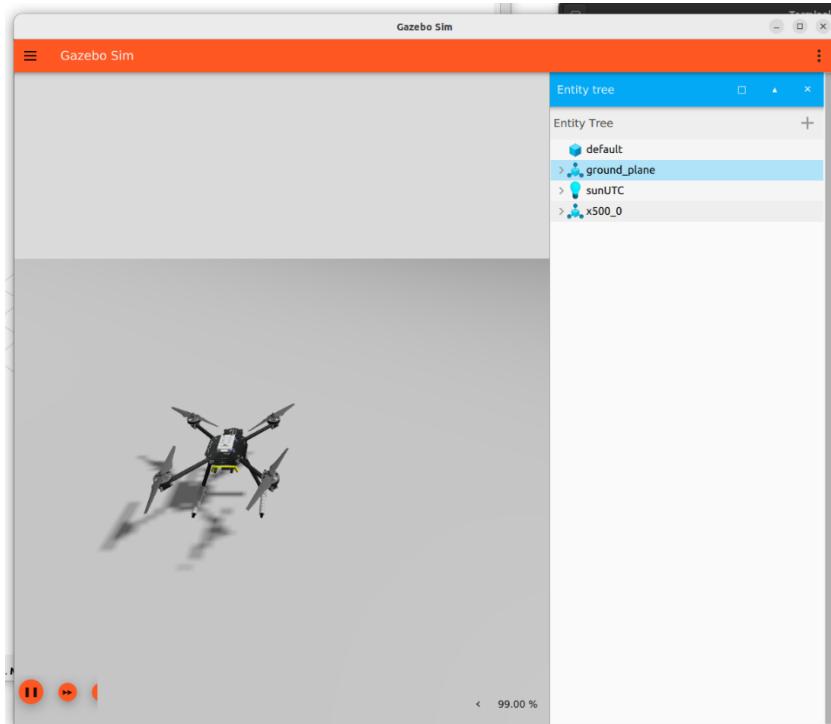
Bước 1: Khởi động Drone: Nhấn space để khởi động drone

```
This node takes keypresses from the keyboard and publishes them
as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

Arm toggle is now: True
X: 0.0    Y: 0.0    Z: 0.0    Yaw: 0.0
```

Ở gazebo sim, drone bắt đầu khởi động quay cánh:



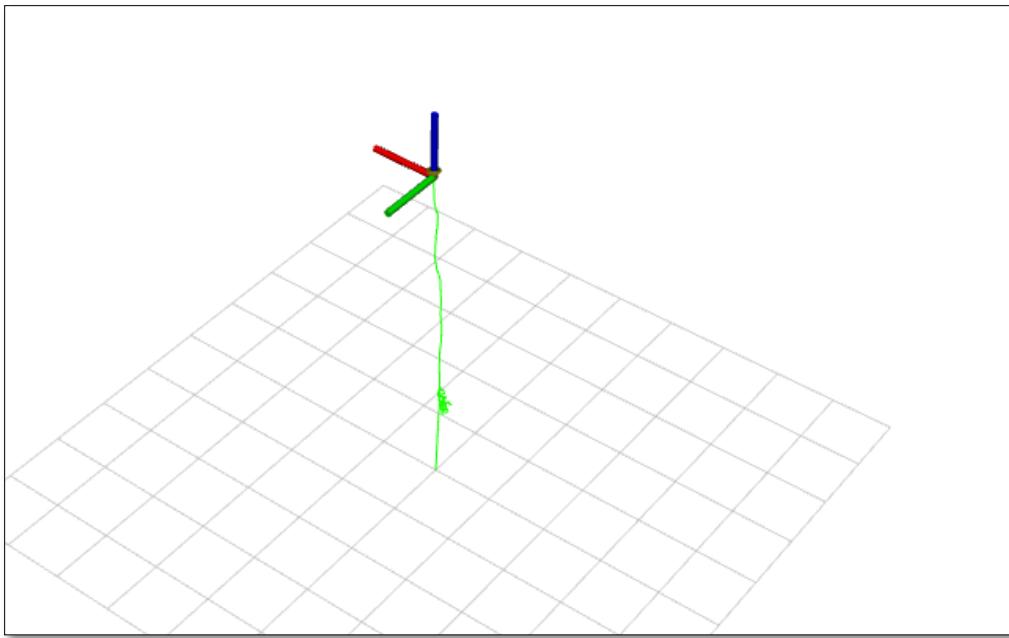
Bước 2: Nhấn “w” 1 lần tức là drone bay theo trục z dương (bay lên) với $v = 0.5 \text{ m/s}$

```
This node takes keypresses from the keyboard and publishes them
as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

Arm toggle is now: True
X: 0.0    Y: 0.0    Z: 0.0    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 0.5    Yaw: 0.0
```

A screenshot of a terminal window titled "Terminal". The window contains text instructions for controlling a drone using keyboard input. It includes mappings for movement (W, S, A, D, arrow keys), pitch and roll (Up/Down, Left/Right arrows), and special functions (Space, T, Y, R). It also displays current sensor values for position and orientation.



Bước 3: Nhấn “w” lần 2 tức là drone bay theo trục z dương (bay lên) với $v= 1\text{m/s}$

```
Terminal

This node takes keypresses from the keyboard and publishes them
as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

Arm toggle is now: True
X: 0.0    Y: 0.0    Z: 0.0    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 0.5    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 1.0    Yaw: 0.0
```

Bước 4: Nhấn “s” lần 1 tức là drone hạ tốc độ lại và vẫn bay theo trục z dương với $v= 0.5\text{m/s}$

```
This node takes keypresses from the keyboard and publishes them
as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

Arm toggle is now: True
X: 0.0      Y: 0.0      Z: 0.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.5      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 1.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.5      Yaw: 0.0
```

Bước 5: Nhấn “s” lần 2 tức là drone hạ tốc độ lại với $v = 0$ m/s theo trục z, tức là drone đang trạng thái lơ lửng vì x:0 y:0 z:0.

```
[+]- Terminal - x
This node takes keypresses from the keyboard and publishes them as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

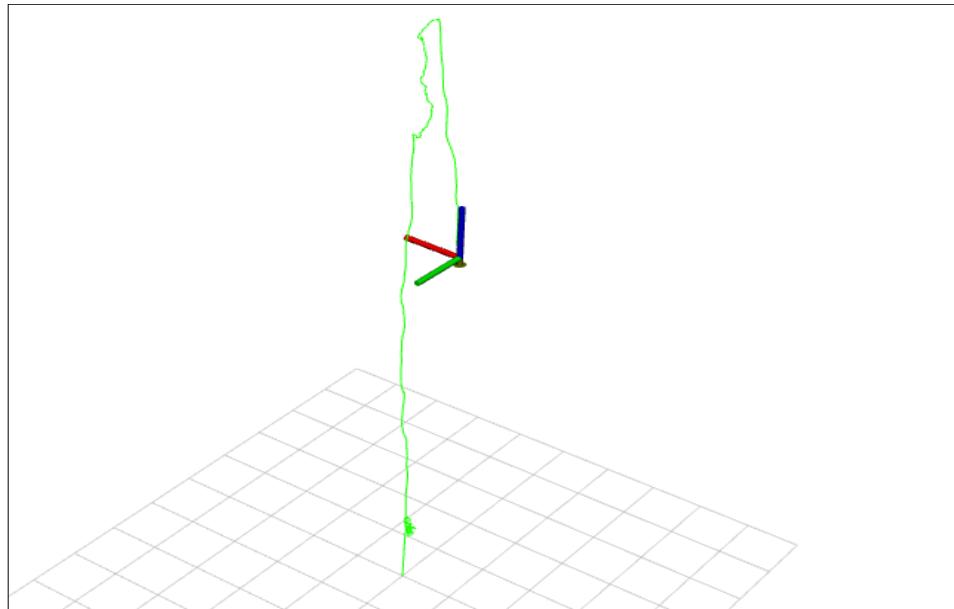
Arm toggle is now: True
X: 0.0    Y: 0.0    Z: 0.0    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 0.5    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 1.0    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 0.5    Yaw: 0.0
X: 0.0    Y: 0.0    Z: 0.0    Yaw: 0.0
```

Bước 6: Nhấn “s” lần 3 lúc này z=-0.5 tức là bay theo trục z âm với tốc độ v=0.5m/s, bây giờ thì drone đang bay xuống.

```
Terminal
as Twist messages.
Using the arrow keys and WASD you have Mode 2 RC controls.
W: Up
S: Down
A: Yaw Left
D: Yaw Right
Up Arrow: Pitch Forward
Down Arrow: Pitch Backward
Left Arrow: Roll Left
Right Arrow: Roll Right

Press SPACE to arm/disarm the drone
Press T to start SINE trajectory (climb to 30m then sine pattern)
Press Y to start CIRCLE trajectory (climb to 30m then circle pattern)
Press R to RESET (return to manual control)

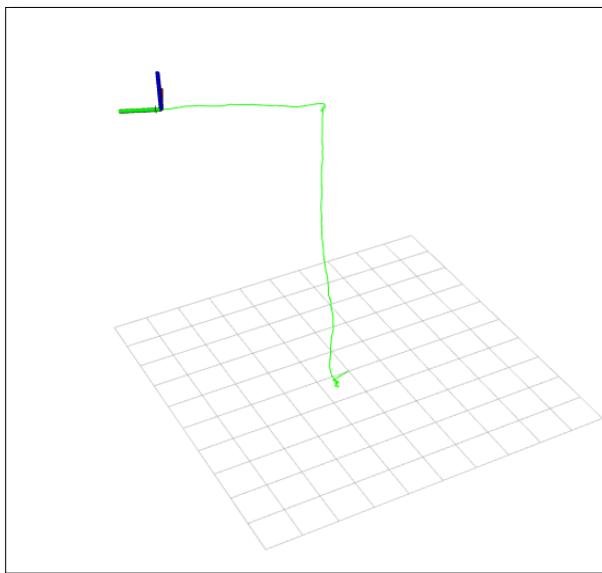
Arm toggle is now: True
X: 0.0      Y: 0.0      Z: 0.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.5      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 1.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.5      Yaw: 0.0
X: 0.0      Y: 0.0      Z: 0.0      Yaw: 0.0
X: 0.0      Y: 0.0      Z: -0.5     Yaw: 0.0
```



- Thực hiện thao tác bay theo trục x:

X: 0.5 Y: 0.0 Z: 0.0 Yaw: 0.0

Bước 1: nhấn nút mũi tên “←” lần 1 thì drone bay theo trục x dương với $v = 0.5 \text{ m/s}$

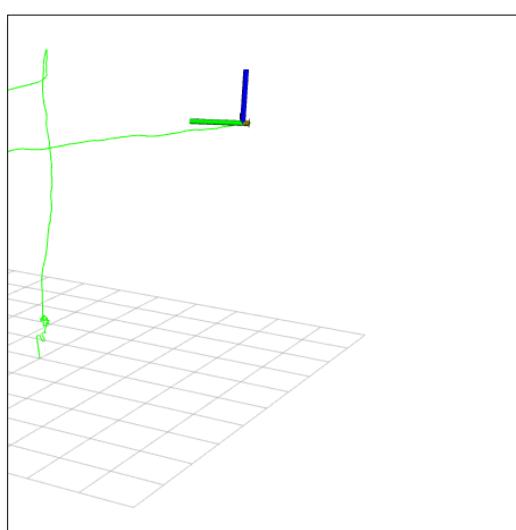


Bước 2: Nhấn nút mũi tên “→” lần 1 thì $x = 0$ drone ở trạng thái đứng yên $v = 0\text{m/s}$

X: 0.0 Y: 0.0 Z: 0.0 Yaw: 0.0

Bước 3: Nhấn nút mũi tên”→” lần 2 thì drone bay theo trục x âm với $v = 0.5\text{m/s}$

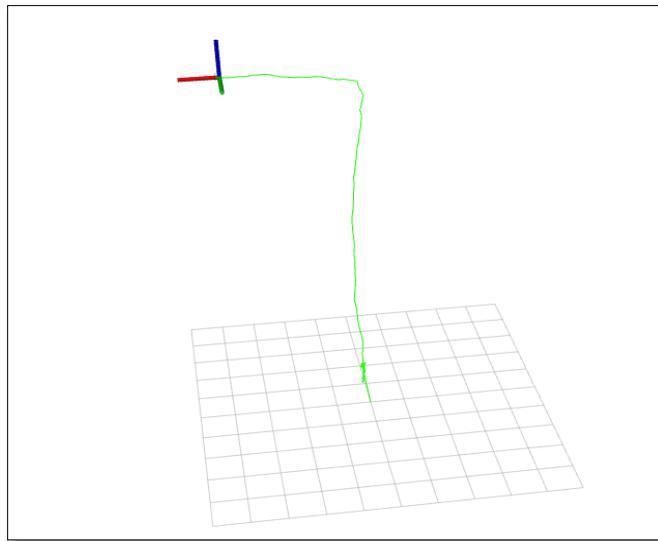
X: -0.5 Y: 0.0 Z: 0.0 Yaw: 0.0



- Thực hiện thao tác bay theo trục y:

Bước 1: nhấn nút “↑” lần 1 thì drone bay theo trục y dương với $v=0.5$ m/s

X: 0.0 Y: 0.5 Z: 0.0 Yaw: 0.0

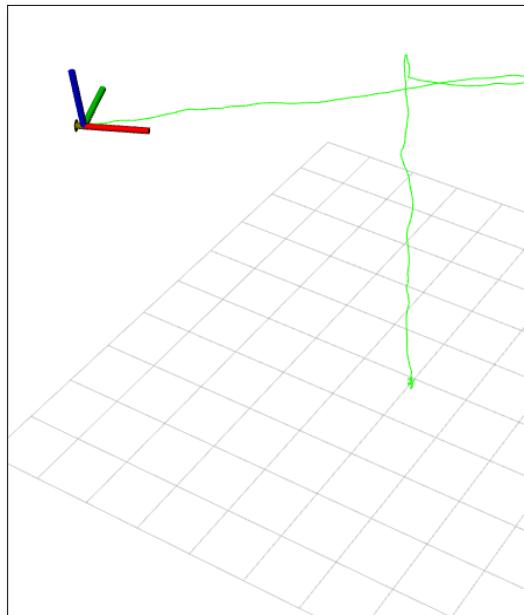


Bước 2: nhấn nút “↓” lần 1 thì $y = 0$ drone ở trạng thái đứng yên với $v=0$ m/s

X: 0.0 Y: 0.0 Z: 0.0 Yaw: 0.0

Bước 3: nhấn nút “↓” lần 2 thì drone bay theo trục y âm với $v=0.5$ m/s

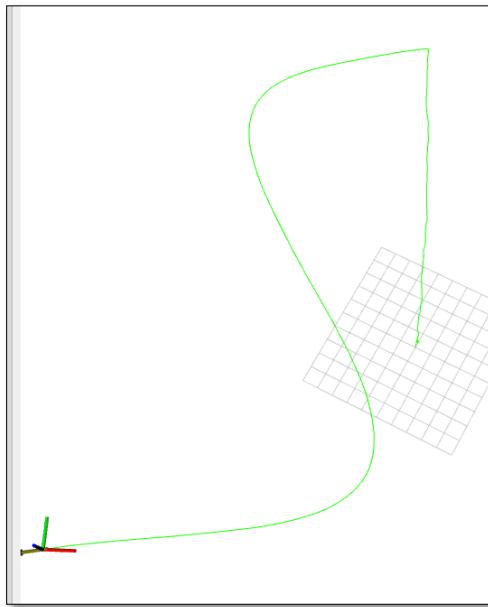
X: 0.0 Y: -0.5 Z: 0.0 Yaw: 0.0



- Thực hiện thao tác bay theo chế độ hình SINE:

Nhấn nút “T” để khởi động chế độ bay hình SINE - drone sẽ bay lên 30m trước khi thực hiện bay theo hình SINE.

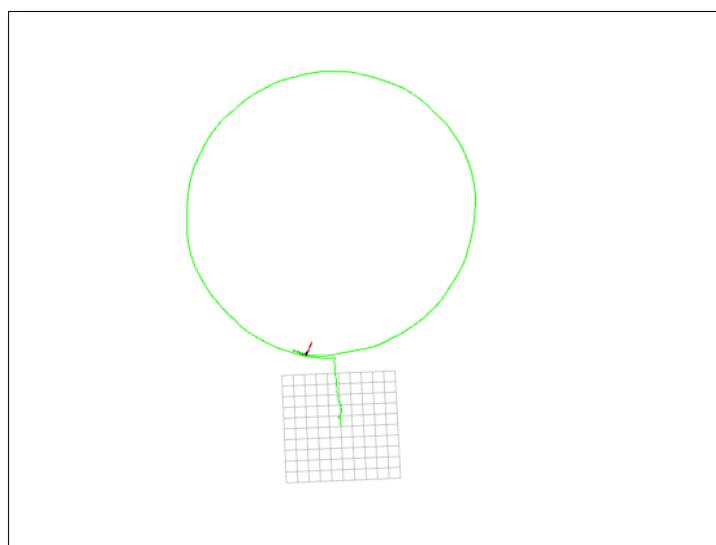
```
Starting SINE trajectory (will climb to 30m first)
Y: 0.0  V: 0.0  Z: 0.0  Yaw: 0.0
```



- Thực hiện thao tác bay theo hình tròn:

Nhấn nút “Y” để khởi động chế độ bay hình tròn - drone sẽ bay lên 30m trước khi thực hiện bay theo hình tròn.

```
Starting CIRCLE trajectory (will climb to 30m first)
```



- Thực hiện thao tác reset để trở lại với thao tác bay theo điều khiển bình thường:

Nhấn nút “R” để reset, tức là có từ bất kỳ chế độ bay hình tròn hoặc sine sẽ trở lại thao tác điều khiển thủ công.

RESET to manual control

3.5. Kết luận chung về cơ chế điều khiển trong PX4 Offboard:

Trong hệ thống điều khiển Offboard sử dụng PX4, ROS2 và Gazebo, cơ chế điều khiển được xây dựng dựa trên việc gửi liên tục các thông điệp TrajectorySetpoint để drone bay đáp ứng theo mong muốn của người vận hành. Các phím điều khiển (W, S, A, D, mũi tên...) không điều khiển drone theo cách tuyệt đối (absolute control), mà theo cơ chế gia tăng vận tốc trên từng trục trong hệ tọa độ ENU. Điều này có nghĩa là mỗi thao tác nhấn phím chỉ làm thay đổi vận tốc mục tiêu, còn độ cao, vị trí hay góc quay được hình thành từ việc drone tích lũy các vận tốc đó theo thời gian.

3.5.1. Điều khiển theo vận tốc (Velocity Control):

Hệ thống hoạt động hoàn toàn dựa trên vận tốc:

- W: tăng vận tốc trục Z → theo trục dương z.
- S: giảm vận tốc trục Z → theo trục âm z.
- ↑ / ↓: điều khiển pitch → bay theo trục dương/âm y.
- ← / →: điều khiển roll → bay theo trục dương/ âm x.
- A/D: điều khiển yaw → quay trái / phải (thay đổi yaw)

Mỗi phím tương ứng với việc thay đổi một giá trị vận tốc theo đơn vị m/s, thay vì điều khiển vị trí hay góc tuyệt đối. Vì vậy, nếu người điều khiển không trả vận tốc về 0, drone sẽ tiếp tục di chuyển mãi theo quán tính của tốc độ đã được thiết lập.

3.5.2. Tính chất của điều khiển Offboard:

- Drone chỉ thực sự “nghe” lệnh khi PX4 đang ở trạng thái Offboard (NAV_STATUS = 14).
- Node điều khiển phải gửi setpoint liên tục mỗi 20 ms.
- Nếu dừng gửi → PX4 tự động thoát Offboard và chuyển sang Loiter để giữ vị trí.

Điều này đảm bảo độ an toàn và tính ổn định trong điều khiển khi kết nối với máy tính bị gián đoạn.

Tài liệu tham khảo

- [1] <https://docs.px4.io/main/en/>
- [2] <https://docs.ros.org/en/humble/index.html>
- [3] https://docs.px4.io/main/en/ros2/offboard_control?utm_