

**ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA**

TRẦN HOÀNG KHÔI NGUYỄN

**HỆ THỐNG ĐỊNH VỊ TÍCH HỢP
THỊ GIÁC LẬP THỂ, QUẢN TÍNH VÀ GPS**

STEREO-VISION/INS/GPS INTEGRATED NAVIGATION SYSTEM

Chuyên ngành : KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA
Mã số : 8520216

LUẬN VĂN THẠC SĨ

TP. HỒ CHÍ MINH, tháng 09 năm 2020

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn đến gia đình. Con cảm ơn Ba, Mẹ, Bà nội đã nuôi nấng con nên người và cho con tình thương vô bờ bến. Cảm ơn Ba đã cho con niềm tin và động lực mạnh mẽ để vượt qua khó khăn. Cảm ơn Mẹ đã là hậu phương vững chắc lo cho con từng bữa ăn giấc ngủ. Cảm ơn em trai đã lo việc nhà cho anh trong những ngày viết luận. Cảm ơn các cô, chú trong gia đình đã tin tưởng và dành cho con những gì tốt nhất.

Em xin gửi lời cảm ơn sâu sắc đến TS. Nguyễn Vĩnh Hảo. Cảm ơn Thầy đã chỉ dạy tận tình và tạo điều kiện tốt nhất cho em trong hai năm vừa qua. Thầy là tấm gương sáng cho em không chỉ về chuyên môn mà còn về nhân cách. Trong công việc, Thầy luôn thẳng thắn chỉ ra cái sai để em tiến bộ. Trong cuộc sống, Thầy luôn gần gũi và quan tâm đến học trò. Được theo Thầy là may mắn hiếm có của em, giúp em tin vào con đường thành công bằng nỗ lực chân chính.

Em xin trân trọng cảm ơn các thầy cô trong Bộ môn Điều Khiển Tự Động, Khoa Điện – Điện tử nói riêng và Trường Đại học Bách khoa nói chung đã truyền dạy cho em kiến thức và tư duy kỹ thuật quý báu mà em sẽ mang theo suốt cuộc đời.

Cuối cùng, xin cảm ơn các anh em trong phòng thí nghiệm 207-B3 đã đồng hành cùng tôi trong công việc và cuộc sống. Cảm ơn Tiến Dũng đã là người bạn đáng quý, cùng tôi vượt qua khó khăn trong những năm tháng vừa qua.

TP.HCM, ngày 19 tháng 08 năm 2020

Thân mến

TRẦN HOÀNG KHÔI NGUYỄN

TÓM TẮT

Chất lượng định vị quyết định không nhỏ đến độ chính xác của robot tự hành. Hệ thống định vị toàn cầu (GPS), mặc dù cho kết quả toàn cục, thường có nhiễu lớn và không khả dụng trong môi trường bị che chắn. Hệ thống định vị thị giác-quán tính (VIO) gần đây nổi lên như một phương pháp định vị cục bộ chính xác cao, tốc độ cập nhật nhanh, ngõ ra 6 trục, hoạt động được ở trong nhà lẫn ngoài trời. Tuy vậy, kết quả VIO bị tích lũy sai số theo thời gian và không phù hợp cho quãng đường di chuyển dài. Luận văn này thực hiện hai nhiệm vụ chính: xây dựng hệ định vị VIO dùng stereo camera theo phương pháp tối ưu hóa phi tuyến; từ đó xây dựng hệ định vị tích hợp VIO/GPS sử dụng bộ lọc Kalman mở rộng nhằm triệt tiêu sai số tích lũy. Phần mềm hệ thống chạy trên máy tính nhúng trong thời gian thực, trên nền tảng hệ điều hành robot (ROS). Nhiều thử nghiệm trong các điều kiện khác nhau cho thấy hệ thống định vị hoạt động ổn định và chính xác. Sản phẩm làm ra có thể ứng dụng cho xe tự lái và các loại robot di động nói chung.

ABSTRACT

Positioning quality determines the accuracy of autonomous robots. Although given the global result, the global positioning system (GPS) contains large noise and is not available in the sheltered environment. The Visual-Inertial Odometry method (VIO) recently emerges as a precise localization method, with fast update speed, 6-axis output, working indoors and outdoors. However, the VIO output accumulates errors over time and is not suitable for long travel distances. This thesis focusses on two main tasks: building a VIO system using stereo camera based on the nonlinear optimization method and thereby building the integrated VIO/GPS navigation system using the extended Kalman filter to eliminate the cumulative error. The system software runs on an embedded computer in real time, based on the robot operating system (ROS). Many tests conducted in different conditions show that the navigation system works stably and accurately. The developed system can be applied to self-driving cars as well as mobile robots.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU	1
1.1. Mở đầu.....	1
1.2. Tổng quan tình hình nghiên cứu.....	2
1.3. Mục tiêu của đề tài.....	4
1.4. Bố cục của luận văn.....	5
CHƯƠNG 2. HỆ THỐNG ĐỊNH VỊ THỊ GIÁC LẬP THỂ.....	6
2.1. Tổng quan hệ thống định vị thị giác lập thể	6
2.2. Giải tam giác dùng hình học epipolar.....	9
2.3. Thuật toán tối ưu hóa phi tuyến Powell's dogleg.....	10
CHƯƠNG 3. HỆ THỐNG ĐỊNH VỊ TÍCH HỢP THỊ GIÁC LẬP THỂ VÀ QUÁN TÍNH	16
3.1. Hiệu chuẩn IMU/StereoCamera	16
3.1.1. Đồng bộ thời gian IMU/Camera.....	16
3.1.2. Hiệu chuẩn không gian IMU/StereoCamera	18
3.2. Thuật toán tích hợp IMU/Camera	22
3.2.1. Trạng thái hệ tích hợp.....	22
3.2.2. Mô hình động học IMU.....	22
3.2.3. Hàm mục tiêu tối ưu hóa	25
3.2.4. Hàm sai số camera.....	26
3.2.5. Hàm sai số IMU	26
CHƯƠNG 4. HỆ THỐNG ĐỊNH VỊ TÍCH HỢP THỊ GIÁC LẬP THỂ, QUÁN TÍNH VÀ GPS	28
4.1. Căn chỉnh hệ thống VIO/GPS	28
4.1.1. Thuật toán căn chỉnh tọa độ	28
4.1.2. Kết quả thực nghiệm căn chỉnh hệ VIO/GPS.....	32
4.2. Thuật toán định vị tích hợp VIO/GPS	34
4.2.1. Bộ lọc Kalman mở rộng EKF.....	34

4.2.2.	Mô hình toán hệ thống định vị tích hợp VIO/GPS.....	36
4.2.3.	Thuật toán định vị tích hợp VIO/GPS sử dụng bộ lọc EKF.....	38
CHƯƠNG 5. THỰC HIỆN VÀ ĐÁNH GIÁ HỆ THỐNG.....		41
5.1.	Thực hiện hệ thống	41
5.1.1.	Phần cứng	41
5.1.2.	Phần mềm	44
5.2.	Phương pháp đánh giá sai số ước lượng.....	49
5.2.1.	Quy đổi tọa độ	49
5.2.2.	Sai số tuyệt đối	50
5.2.3.	Sai số tương đối.....	50
5.3.	Kết quả thử nghiệm trong nhà	52
5.3.1.	Thử nghiệm với thanh trượt tuyến tính	52
5.3.2.	Thử nghiệm với người đi bộ.....	54
5.4.	Kết quả thử nghiệm ngoài trời.....	56
5.4.1.	Thử nghiệm trên sân trường	58
5.4.2.	Thử nghiệm trên đường phố.....	67
5.4.3.	Những trường hợp thị giác suy giảm và hướng khắc phục	71
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		74
6.1.	Kết luận.....	74
6.2.	Hướng phát triển.....	74
TÀI LIỆU THAM KHẢO		75
DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ CỦA TÁC GIẢ.....		79

DANH MỤC HÌNH ẢNH

Hình 1.1. Hệ thống định vị trên xe hơi tự lái Tesla	1
Hình 1.2. Flycam Mark sử dụng hệ thống định vị tích hợp thị giác quán tính	2
Hình 1.3. Kết quả của nghiên cứu OKVIS [8]. Thuật toán (aslam) đạt sai số vị trí 2% trên tổng quãng đường di chuyển.....	3
Hình 2.1. Sơ đồ tổng quan hệ thống định vị dùng thị giác lập thể.....	6
Hình 2.2. Matching cặp ảnh hiện tại với cặp ảnh keyframe gần nhất.....	7
Hình 2.3. Một số lượng keyframe và khung hình gần nhất được duy trì.....	8
Hình 2.4. Mô hình pinhole camera	9
Hình 2.5. Hình học epipolar.....	10
Hình 2.6. Bước thay đổi tổng hợp trong thuật toán Powell's dogleg	14
Hình 3.1. Giảm đồ đồng bộ thời gian IMU/Camera	17
Hình 3.2. Các hệ tọa độ của hệ thống IMU/StereoCamera và mẫu hiệu chuẩn.....	19
Hình 3.3. Thông số méo dạng equidistant camera	20
Hình 3.4. Hệ stereo camera tìm được từ quá trình hiệu chuẩn	20
Hình 3.5. Phân bố sai số reprojection trên camera trái và phải trong bước hiệu chuẩn camera-to-IMU	21
Hình 3.6. Giảm đồ lấy mẫu IMU và Camera	26
Hình 4.1 Quỹ đạo VIO, GPS ban đầu và quỹ đạo chuẩn RTK.....	32
Hình 4.2 Các quỹ đạo sau khi căn chỉnh.....	33
Hình 4.3 Sai số căn chỉnh so với quỹ đạo chuẩn theo thời gian	34
Hình 5.1. Stereo camera Point Grey Bumblebee2	41
Hình 5.2. IMU ADIS16488 và mạch xử lý	41
Hình 5.3. GPS Receiver Ublox NEO-M8N	42
Hình 5.4. Máy tính nhúng IEI NANO-HM650.....	42
Hình 5.5. GPS RTK South S82T rover và base	42
Hình 5.6. Novatel SPAN-CPT	43

Hình 5.7. Sơ đồ phần cứng hệ thống StereoVision/INS/GPS và các thiết bị đánh giá	43
Hình 5.8. Sơ đồ phần cứng đồng bộ IMU-Camera	43
Hình 5.9. Mô hình phần cứng hệ thống StereoVision/INS/GPS	44
Hình 5.10. Cơ chế truyền nhận bản tin trong ROS	45
Hình 5.11. File dữ liệu bag.....	45
Hình 5.12. Môi trường ảo rviz	46
Hình 5.13. Cấu trúc phần mềm lớp Driver.....	47
Hình 5.14. Cấu trúc phần mềm lớp Process.....	48
Hình 5.15. Cấu trúc phần mềm lớp Display.....	48
Hình 5.16. Quy đổi tọa độ giữa bộ ước lượng và bộ chuẩn.....	49
Hình 5.17. Sai số tương đối giữa quỹ đạo ước lượng và quỹ đạo đúng.....	51
Hình 5.18. Dạng đồ thị boxplot.....	52
Hình 5.19. Thanh trượt dùng trong thử nghiệm.....	53
Hình 5.20. Ảnh trái và phải từ stereo camera trong tập dữ liệu Indoor-Slider	53
Hình 5.21. Vị trí ước lượng hệ VIO, tập dữ liệu Indoor-Slider	54
Hình 5.22. Sai số ước lượng vị trí và góc yaw, tập dữ liệu Indoor-Slider	54
Hình 5.23. Vị trí ước lượng VIO, tập dữ liệu Indoor-Walking.....	55
Hình 5.24. Sai số ước lượng vị trí và góc yaw, tập dữ liệu Indoor-Walking.....	55
Hình 5.25. Ảnh tái dựng căn phòng từ các landmark thừa thớt và quỹ đạo ước lượng trong thử nghiệm Indoor-Walking	56
Hình 5.26. Bố trí thử nghiệm ngoài trời.....	57
Hình 5.27. Ảnh stereo camera tại điểm xuất phát trong thử nghiệm trên sân trường	57
Hình 5.28. Kết quả định vị vị trí, tập dữ liệu Outdoor-Circle.....	58
Hình 5.29. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Circle.....	59
Hình 5.30. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Circle	59
Hình 5.31. Sai số tương đối vị trí và góc xoay theo quãng đường, tập dữ liệu Outdoor-Circle	60

Hình 5.32. Kết quả định vị vị trí, tập dữ liệu Outdoor-Square	61
Hình 5.33. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Square.....	62
Hình 5.34. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Square.....	62
Hình 5.35. Thống kê sai số vị trí và góc xoay tương đối theo quãng đường, tập dữ liệu Outdoor-Square	63
Hình 5.36. Kết quả định vị vị trí, tập dữ liệu Outdoor-Rectangle	64
Hình 5.37. Sai số vị trí theo thời gian (trái) và sai số vị trí tương đối theo quãng đường (phải), tập dữ liệu Outdoor- Rectangle	65
Hình 5.38. Kết quả tự hiệu chỉnh vị trí 1, $x_0 = y_0 = z_0 = 10\text{m}$, $\psi_0 = 20^\circ$	66
Hình 5.39. Kết quả tự hiệu chỉnh vị trí 2, $x_0 = y_0 = z_0 = 30\text{m}$, $\psi_0 = 60^\circ$	66
Hình 5.40. Kết quả tự hiệu chỉnh vị trí 3, $x_0 = y_0 = z_0 = 100\text{m}$, $\psi_0 = 180^\circ$	66
Hình 5.41. Đoạn đường Lý Thường Kiệt trong thử nghiệm.....	68
Hình 5.42. Đoạn đường Thành Thái trong thử nghiệm	68
Hình 5.43. Đoạn đường Tô Hiến Thành trong thử nghiệm	68
Hình 5.44. Kết quả định vị vị trí, tập dữ liệu Outdoor-Street	69
Hình 5.45. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Street	69
Hình 5.46. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Street.....	70
Hình 5.47. Sai số tương đối vị trí và góc xoay theo quãng đường, tập dữ liệu Outdoor-Street.....	70
Hình 5.48. Trường hợp ảnh bị chói sáng do mặt trời chiếu trực tiếp hoặc gián tiếp	72
Hình 5.49. Trường hợp đa số vật thể trong tầm nhìn camera chuyển động đều.....	73

DANH MỤC BẢNG BIỂU

Bảng 2.1. Tóm tắt thuật toán Powell's dogleg.....	14
Bảng 3.1. Thông số mô hình chiếu pinhole camera.....	20
Bảng 3.2. Thông số méo dạng equidistant camera	20
Bảng 3.3. Sai số hiệu chuẩn camera-to-IMU	21
Bảng 4.1. Tóm tắt thuật toán căn chỉnh với 1 góc xoay yaw.....	31
Bảng 4.2 Kết quả tọa độ và sai số căn chỉnh.....	33
Bảng 4.3. Bộ lọc EKF rời rạc.....	35
Bảng 5.1. Sai số tuyệt đối vị trí, tập dữ liệu Outdoor-Circle	60
Bảng 5.2. Sai số tuyệt đối góc xoay, tập dữ liệu Outdoor-Circle	60
Bảng 5.3. Sai số vị trí tuyệt đối, tập dữ liệu Outdoor-Square.....	63
Bảng 5.4. Sai số góc xoay tuyệt đối, tập dữ liệu Outdoor-Square	63
Bảng 5.5. Sai số RMS vị trí, tập dữ liệu Outdoor-Rectangle.....	65
Bảng 5.6. Thời gian tự hiệu chỉnh của hệ VIO/GPS.....	67
Bảng 5.7. Sai số RMS vị trí, tập dữ liệu Outdoor-Street	71
Bảng 5.8. Sai số RMS góc xoay, tập dữ liệu Outdoor-Street	71

DANH MỤC TỪ VIẾT TẮT

GPS	Global Positioning System	Hệ thống định vị toàn cầu
IMU	Inertial Measurement Unit	Đơn vị đo lường quán tính
INS	Inertial Navigation System	Hệ thống định vị quán tính
VO	Visual Odometry	Hệ thống đo lường thị giác
VIO	Visual Inertial Odometry	Hệ thống đo lường thị giác quán tính
ENU	East North Up	Hệ tọa độ Đông-Bắc-Lên
NED	North East Down	Hệ tọa độ Bắc-Đông-Xuống
RTK	Real-time Kinematic	Hệ đo động thời gian thực
ROS	Robot Operating System	Hệ điều hành robot

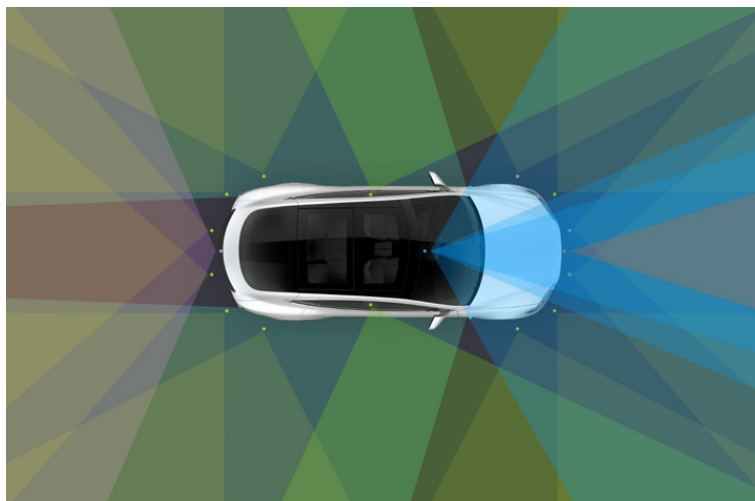
DANH MỤC KÍ HIỆU TOÁN

a	Lượng vô hướng
$\mathbf{a}$	Vector
$\mathbf{A}$	Ma trận
$f(a)$	Hàm số của a
$\bar{\mathbf{x}}$	Giá trị điểm làm việc của $\mathbf{x}$
$\tilde{\mathbf{x}}$	Giá trị đo của $\mathbf{x}$
$\hat{\mathbf{x}}$	Giá trị ước lượng hậu nghiệm (posterior) của $\mathbf{x}$
$\hat{\mathbf{x}}^-$	Giá trị ước lượng tiên nghiệm (prior) của $\mathbf{x}$
$\mathbf{x}_k$	Vector $\mathbf{x}$ tại thời điểm k
$\delta\mathbf{x}$	Số gia (perturbation) của $\mathbf{x}$ giữa thời điểm k và $k - 1$
$\mathcal{F}_A$	Hệ tọa độ
${}_A\mathbf{v}$	Vector $\mathbf{v}$ biểu diễn trong hệ tọa độ $\mathcal{F}_A$
$\mathbf{t}_{AB}$	Vector tịnh tiến từ $\mathcal{F}_A$ đến $\mathcal{F}_B$
$\mathbf{p}_{AB}$	Vector vị trí của $\mathcal{F}_B$ trong $\mathcal{F}_A$
$\mathbf{R}_{AB}$	Ma trận xoay từ $\mathcal{F}_A$ đến $\mathcal{F}_B$
$\mathbf{T}_{AB}$	Ma trận biến đổi tọa độ thuần nhất từ $\mathcal{F}_A$ đến $\mathcal{F}_B$
$[\cdot]^\times$	Toán tử tích chéo

CHƯƠNG 1. GIỚI THIỆU

1.1. Mở đầu

Định vị là khả năng thiết yếu của robot khi di chuyển trong môi trường. Thông tin định vị chính xác là yêu cầu đầu tiên để robot chuyển động đúng. Ví dụ trên xe tự lái, dữ liệu vị trí và góc hướng cho phép xe bám theo quỹ đạo mong muốn. Trên máy bay hoặc drone, dữ liệu góc nghiêng và vị trí giúp phương tiện giữ cân bằng trên không. Trong những công nghệ định vị hiện nay, hệ thống định vị thị giác Visual Odometry – VO) đang được quan tâm nghiên cứu nhờ vào giá thành phải chăng, độ chính xác cao, không lệ thuộc vào cơ cấu cơ khí của phương tiện, hoạt động được ở môi trường trong nhà cũng như ngoài trời. Đơn cử là hệ thống định vị trên xe hơi tự lái của hãng Tesla sử dụng 7 camera kết hợp với cảm biến radar và siêu âm [1].



Hình 1.1. Hệ thống định vị trên xe hơi tự lái Tesla

Hệ định vị tích hợp thị giác quán tính (Visual Inertial Odometry – VIO) sử dụng thông tin quán tính để bổ sung cho thị giác, mang đến nhiều lợi ích. Trên máy bay, khả năng đo góc nghiêng chuẩn và tốc độ cập nhật nhanh của IMU cho phép máy bay giữ ổn định tốt và di chuyển với tốc độ cao. Quán tính còn đóng vai trò trợ giúp thị giác trong nhiều trường hợp suy giảm như môi trường ít đặc trưng, chuyển động xoay gắt. Ngược lại, thông tin thị giác khi tốt sẽ góp phần ước lượng đúng bias IMU, từ đó

nâng chất lượng hệ tích hợp. Điển hình là sản phẩm flycam Mark ứng dụng hệ định vị thị giác quán tính để tự động bám theo vị trí người dùng [2].



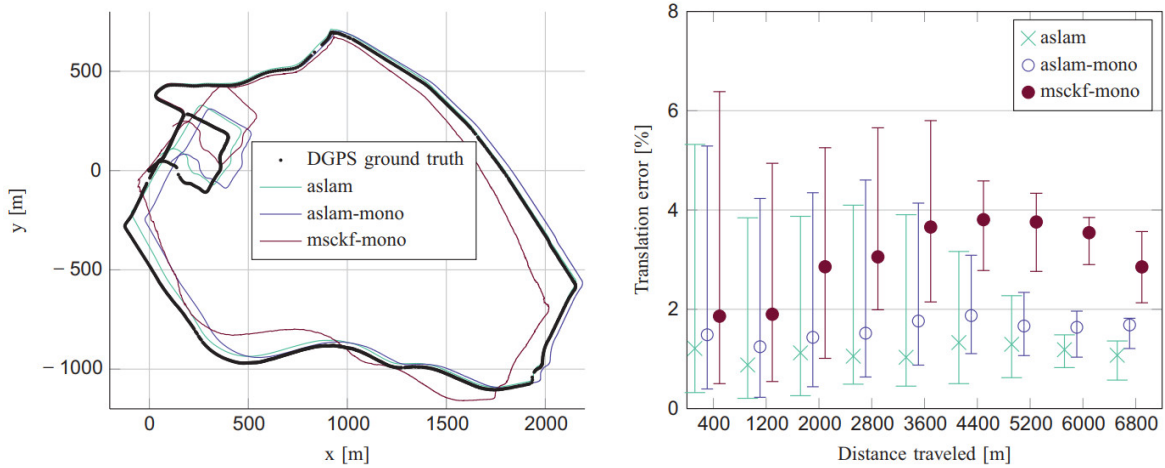
Hình 1.2. Flycam Mark sử dụng hệ thống định vị tích hợp thị giác quán tính

Tuy sở hữu nhiều ưu điểm, hệ VIO không tránh khỏi sai số tích lũy theo thời gian vì bản chất là một phương pháp định vị tương đối. Khi quãng đường di chuyển tăng lên, kết quả định vị VIO sẽ trôi ngày càng xa giá trị đúng. Để khử độ trôi của hệ VIO, giải pháp là tích hợp với một hệ định vị tuyệt đối như hệ thống định vị toàn cầu (Global Positioning System – GPS), giúp cung cấp vị trí toàn cục không trôi của vật thể trên Trái Đất. Ở chiều ngược lại, VIO hỗ trợ giảm nhiễu cục bộ, cũng như trợ giúp GPS trong môi trường có nhiều vật cản như cây cối, nhà cao tầng. Hệ thống định tích hợp VIO/GPS có khả năng cung cấp ngõ ra đầy đủ 6 trục với tốc độ cập nhật nhanh, độ chính xác cao, không trôi theo thời gian và bền vững với điều kiện môi trường. Đây cũng là mục tiêu mà đề tài hướng đến.

1.2. Tổng quan tình hình nghiên cứu

Các nghiên cứu về hệ VIO có thể chia thành hai nhóm: dựa trên bộ lọc Kalman và dựa trên tối ưu hóa phi tuyến. Nhóm thứ nhất được phát triển đa dạng với bộ lọc Kalman mở rộng – EKF [3], bộ lọc Kalman mở rộng lặp – IEKF [4] và bộ lọc Kalman phi tập trung – UKF [5]. Ưu điểm của bộ lọc Kalman là quy trình tính toán đệ quy gọn nhẹ, có khả năng chạy trên các thiết bị di động [6]. Mặt khác, nhóm dựa trên tối ưu hóa phi tuyến bắt nguồn từ bài toán định vị thị giác, trong đó tọa độ camera được

ước lượng từ rất nhiều điểm đặc trưng sao cho cực tiểu hàm tổng bình phương sai số [7]. Khi dùng vào hệ tích hợp, hàm sai số không chỉ được thêm thành phần quán tính, mà còn mở rộng về các trạng thái quá khứ trong một cửa sổ giới hạn [8] [9]. Việc xét các yếu tố trong quá khứ giúp cho kết quả ước lượng phản ánh đối tượng một cách bao quát hơn. Bên cạnh đó, nhờ đặc tính thừa thớt của bài toán, phương pháp tối ưu hóa được chứng minh là có độ chính xác cao hơn cho cùng một khối lượng tính toán khi so sánh với phương pháp lọc [10].



Hình 1.3. Kết quả của nghiên cứu OKVIS [8]. Thuật toán (aslam) đạt sai số vị trí 2% trên tổng quãng đường di chuyển.

Xét về mức độ tích hợp, hệ định vị có thể chia thành tích hợp lỏng (loosely coupled) và tích hợp chặt (tightly coupled). Trong hệ tích hợp lỏng, mỗi cảm biến thực hiện ước lượng riêng rẽ và chỉ kết hợp dữ liệu với nhau ở giai đoạn cuối. Ưu điểm của phương pháp này là thiết kế đơn giản, dễ dàng kết hợp các module có sẵn mà không phải thay đổi hoạt động bên trong. Ví dụ, nghiên cứu [11] kết hợp dữ liệu định vị thị giác từ thuật toán SVO [12] với giá trị ước lượng IMU thông qua bộ tích hợp lỏng MSF [13]. Nền tảng điều khiển quadrotor PIXHAWK [14] thực hiện tích hợp lỏng thị giác và IMU bằng bộ lọc EKF. Tuy nhiên, việc chưa xét đến mối liên hệ bên trong giữa các cảm biến làm cho hệ tích hợp lỏng không đạt được độ chính xác cao. Những nghiên cứu gần đây sử dụng phương pháp tích hợp chặt, theo đó dữ liệu được chia sẻ dựa trên ràng buộc giữa các cảm biến nhằm đạt được kết quả ước lượng tối ưu. Với các hệ tích hợp chặt dựa trên bộ lọc như ROVIO [4], MSCKF [15], giá trị

đo IMU được dùng để dự báo vector trạng thái, trong khi giá trị đo camera được dùng để cập nhật. Mặt khác, với hệ tích hợp chặt theo phương pháp tối ưu hóa như OKVIS [8], sai số IMU và camera được ước lượng đồng thời theo hàm chi phí tổng hợp. Kết quả so sánh tại [16] cho thấy các hệ tích hợp chặt thị giác – quán tính thể hiện độ chính xác vượt trội so với hệ tích hợp lỏng trong cùng một điều kiện. Từ các nhận xét trên, học viên chọn xây dựng hệ định vị VIO trong đề tài theo mô hình tích hợp chặt sử dụng phương pháp tối ưu hóa phi tuyến.

Để thực hiện hệ định vị tích hợp VIO và GPS, đề tài chọn mô hình tích hợp lỏng, trong đó mỗi hệ thống đầu vào được xem như một module hoạt động độc lập. Do mục tiêu khử trôi trên kết quả VIO có thể đạt được bằng cách chỉ sử dụng ngõ ra không trôi của GPS, việc can thiệp vào giải thuật bên trong của GPS là chưa thực sự cần thiết. Hơn nữa, vì bản thân hệ VIO đã ước lượng bias của IMU bằng tích hợp chặt, ta có thể bỏ qua việc ước lượng thông số nội IMU trong bước này. Giải thuật ước lượng được chọn là bộ lọc Kalman mở rộng - EKF nhờ hoạt động hiệu quả và dùng ít tài nguyên, cho phép chương trình chạy trong thời gian thực. Việc cập nhật VIO và GPS diễn ra bất đồng bộ theo tần số lấy mẫu riêng của mỗi hệ thống, với giá trị dự đoán VIO được nội suy đến thời điểm cập nhật GPS để giảm thiểu sai số do độ trễ giữa hai hệ định vị.

1.3. Mục tiêu của đề tài

Đề tài có những nhiệm vụ như sau:

- Xây dựng thuật toán định vị tích hợp chặt thị giác lập thể và quán tính sử dụng phương pháp tối ưu hóa phi tuyến.
- Xây dựng thuật toán định vị tích hợp hệ thống thị giác-quán tính với GPS dựa trên bộ lọc Kalman mở rộng.
- Xây dựng hệ thống phần cứng kết hợp Stereo Camera, IMU, GPS; phần mềm xử lý thuật toán VIO/GPS trong thời gian thực trên nền tảng ROS.
- Thu thập dữ liệu, thử nghiệm, đánh giá chất lượng hệ định vị trong nhiều điều kiện và môi trường khác nhau.

Mục tiêu đầu ra của đề tài:

- Sai số tương đối của giải thuật VIO trong môi trường tĩnh là 1% cho vị trí và $0.05^\circ/\text{m}$ cho góc xoay.
- Sai số tuyệt đối của giải thuật tích hợp VIO/GPS trong môi trường tĩnh dưới 1m cho vị trí và dưới 2° cho góc xoay.
- Tốc độ cập nhật tối thiểu 100Hz.

Môi trường tĩnh đề cập ở đây không nhất thiết phải tĩnh tuyệt đối, tuy nhiên thành phần tĩnh cần lớn hơn 50% diện tích môi trường để thuật toán hoạt động đúng.

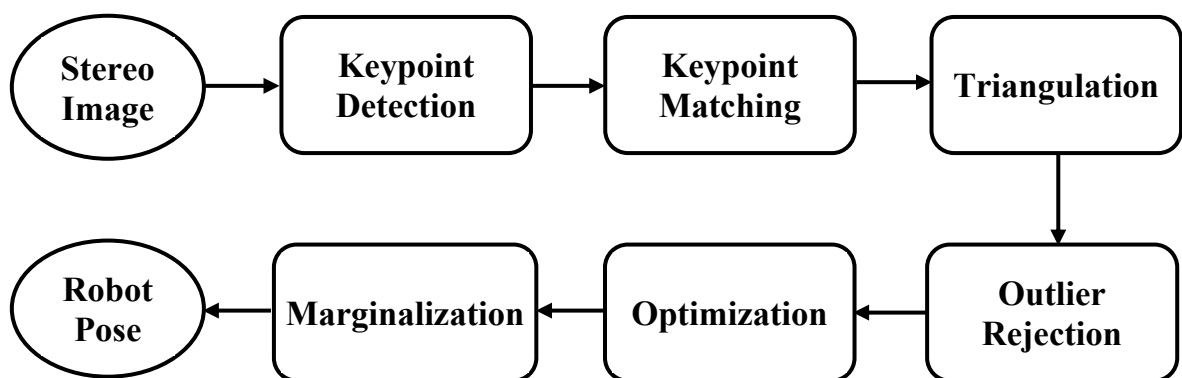
1.4. Bố cục của luận văn

Trong chương 2, học viên trình bày cơ sở lý thuyết về phương pháp định vị dùng thị giác lập thể và giải thuật tối ưu hóa phi tuyến. Chương 3 trình bày phương pháp hiệu chuẩn IMU/Camera và triển khai giải thuật tích hợp chặt dùng tối ưu hóa phi tuyến. Chương 4 trình bày phương pháp căn chỉnh hệ VIO/GPS và triển khai thuật toán tích hợp lỏng dựa trên bộ lọc Kalman mở rộng. Chương 5 mô tả quá trình thực hiện hệ thống và kết quả thử nghiệm trên các tập dữ liệu khác nhau. Cuối cùng, chương 6 đưa ra kết luận và hướng phát triển của đề tài.

CHƯƠNG 2. HỆ THỐNG ĐỊNH VỊ THỊ GIÁC LẬP THỂ

2.1. Tổng quan hệ thống định vị thị giác lập thể

Định vị thị giác (visual odometry - VO) là phương pháp ước lượng chuyển động của robot bằng cách quan sát sự thay đổi của các điểm đặc trưng (keypoint) trong môi trường trên ảnh chụp từ camera, giả sử rằng môi trường là cố định. Hệ thống định vị thị giác lập thể trong đề tài được thực hiện theo 6 giai đoạn như sau:



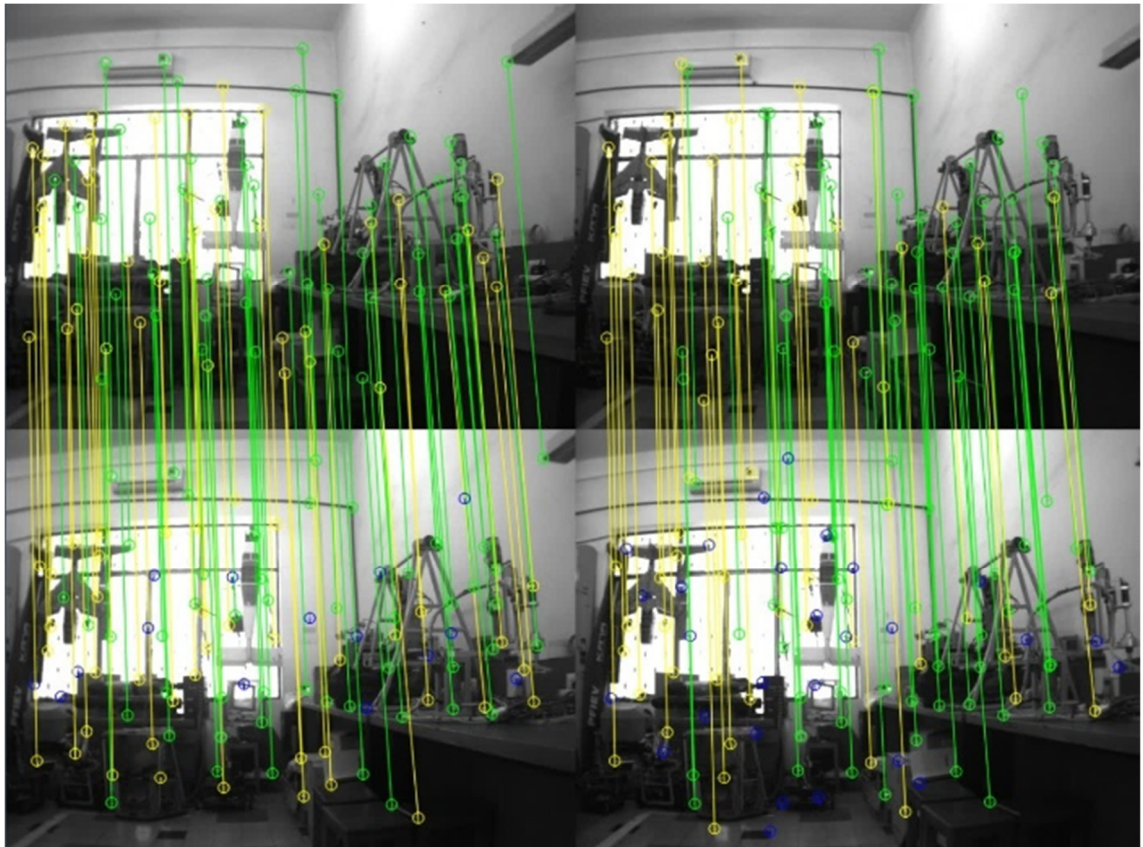
Hình 2.1. Sơ đồ tổng quan hệ thống định vị dùng thị giác lập thể

Keypoint detection and description: phát hiện và mô tả các điểm đặc trưng (keypoint) trong ảnh chụp từ camera. Đây là một kỹ thuật thiết yếu trong xử lý ảnh để phát hiện những điểm ảnh nổi bật và có khả năng lặp lại dưới các góc chụp khác nhau, ví dụ như các góc, cạnh, đốm,... Mỗi điểm đặc trưng được mô tả theo mối tương quan của các pixel xung quanh nó, nhằm tạo ra một tiêu chuẩn để phân biệt với những điểm đặc trưng khác. Theo đó, đề tài dùng bộ phát hiện góc Harris [17] và thuật toán mô tả đặc trưng BRISK [18] để thực hiện giai đoạn này. BRISK cho phép tính toán nhanh hơn các thuật toán trước đây như SIFT và SURF, cũng như đảm bảo độ chính xác lặp lại cao.

Keypoint matching: bắt cặp giữa những điểm đặc trưng có mô tả tương đồng nhau giữa các ảnh. Quá trình matching đi qua hai bước: matching 3D-2D và matching 2D-2D. Bước matching 3D-2D bắt cặp các điểm đặc trưng trong ảnh hiện tại với các landmark 3D theo mô tả BRISK, giả sử rằng một bản đồ landmark 3D cục bộ được

duy trì bởi một số khung hình gần nhất (most current frames) và một số khung hình quan trọng (keyframes) trong quá khứ. Với mỗi cặp trùng khớp, ta gán chỉ số của landmark trong không gian 3D cho keypoint trong tọa độ ảnh 2D. Tiếp theo, bước *matching 2D-2D* được thực hiện để liên hệ những keypoint không có sẵn landmark 3D tương ứng. Ta bắt cặp các keypoint giữa hai ảnh stereo ảnh trái và phải, đồng thời bắt cặp chúng với những cặp ảnh stereo trong quá khứ.

Hình 2.2 ví dụ kết quả matching giữa cặp ảnh hiện tại (dưới) với cặp ảnh keyframe gần nhất (trên). Các đường màu xanh lá thể hiện các điểm đặc trưng tương đồng 3D-2D, trong khi các đường màu vàng thể hiện các điểm tương đồng 2D-2D. Các chấm xanh dương thể hiện hai điểm chỉ tương đồng trong cặp ảnh stereo.

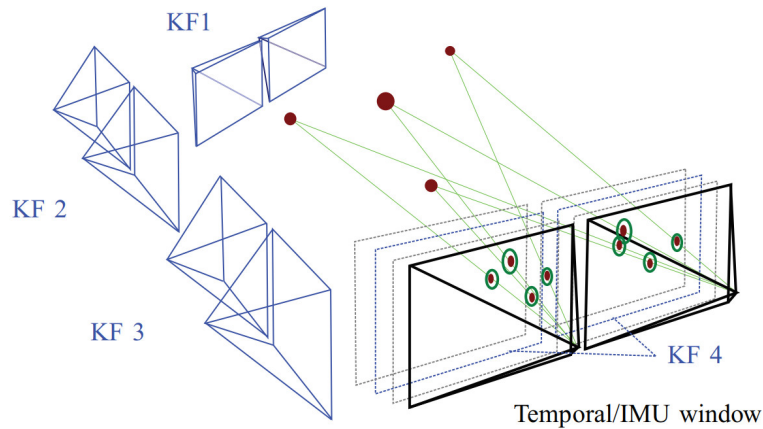


Hình 2.2. Matching cặp ảnh hiện tại với cặp ảnh keyframe gần nhất

Triangulation: giải tam giác mỗi keypoint 2D tương đồng giữa từng cặp ảnh để tính toán độ sâu của landmark. Ta chỉ chọn những landmark có độ sâu ổn định qua nhiều khung hình để đưa vào bản đồ 3D và cấp chỉ số mới. Những keypoint không đưa vào bản đồ được giữ lại cho lần matching tiếp theo.

Outlier rejection: loại bỏ các kết quả ngoại lệ (outlier) trong tập dữ liệu. Outlier xuất hiện do nhiều nguyên nhân: các keypoint được trích từ các vật thể di chuyển trong môi trường, mô tả của keypoint bị sai lệch do điều kiện ánh sáng thay đổi. Việc loại outlier được thực hiện trong bước matching 3D-2D bằng cách xét khoảng cách của keypoint 2D giữa các khung hình và áp dụng thuật toán RANSAC [19]. Trong bước matching 2D-2D, RANSAC được sử dụng giữa ảnh hiện tại và ảnh keyframe gần nhất. Sau bước này, ta đã khởi tạo được tập landmark 3D khá tốt để đưa vào ước lượng.

Optimization: ước lượng đồng thời chuyển động của camera và vị trí landmark từ tập landmark khởi tạo sử dụng phương pháp tối ưu hóa phi tuyến. Thuật toán tối ưu hóa phi tuyến được trình bày cụ thể tại phần 2.3.



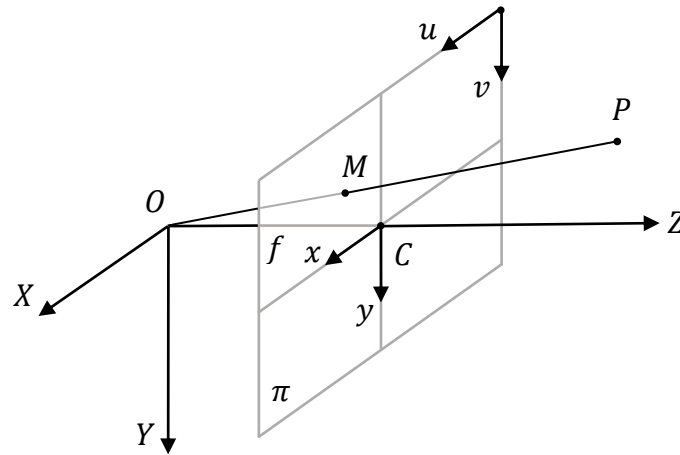
Hình 2.3. Một số lượng keyframe và khung hình gần nhất được duy trì

Marginalization: loại bỏ dữ liệu không cần thiết trong quá khứ. Để matching và ước lượng đủ nhanh, ta chỉ nên duy trì một số lượng giới hạn các khung hình và landmark quá khứ. Như Hình 2.3, các khung hình được giữ lại gồm $S = 4$ khung hình gần nhất và $M = 3$ keyframe ở xa hơn trong quá khứ. Ta chọn keyframe theo chiến lược heuristic. Một khung hình là keyframe khi thỏa một trong hai điều kiện sau: đường bao của tất cả landmark được chiếu và bắt cặp chiếm tỉ lệ diện tích trên ảnh là nhỏ hơn một ngưỡng (thường là 50%); hoặc số tỉ lệ số keypoint được bắt cặp trên số keypoint phát hiện là nhỏ (dưới 20%). Các điều kiện trên nhằm lưu giữ những thông tin quan trọng, mới lần đầu xuất hiện trong ảnh.

2.2. Giải tam giác dùng hình học epipolar

Sau quá trình matching, giả sử ta biết tọa độ 2D của một điểm đặc trưng trên hai ảnh stereo, mục tiêu của phép giải tam giác (triangulation) là tái tạo lại tọa độ 3D của điểm đó trong môi trường.

Trước tiên, ta thành lập mô hình chiếu của camera đơn, thường gọi là mô hình pinhole camera. Khi camera chụp ảnh, không gian 3D trong thời điểm hiện tại được chiếu lên mặt phẳng ảnh 2D. Hình 2.4 mô tả phép chiếu một điểm đặc trưng $P = [X \ Y \ Z]^T \in \mathbb{R}^3$ thành điểm ảnh $M = [x \ y]^T \in \mathbb{R}^2$. Trong đó, XYZ là hệ tọa độ không gian đặt tại tâm quang học camera O , xy là hệ tọa độ mặt phẳng ảnh đặt tại tâm ảnh C . Mặt khác, uv là hệ tọa độ mặt phẳng ảnh đặt tại mép ảnh và có đơn vị là pixel. Tiêu cự camera có độ dài $f = OC$.



Hình 2.4. Mô hình pinhole camera

Bằng các tam giác đồng dạng, ta có mô hình pinhole camera:

$$\frac{X}{x} = \frac{Y}{y} = \frac{Z}{f} = s \quad (2.1)$$

trong đó s được gọi là hệ số tỉ lệ (scale factor) giữa không gian và mặt phẳng ảnh. Gọi k là tỉ lệ pixel/mét, (c_u, c_v) là tọa độ tâm ảnh, ta có thể quy đổi giữa hệ tọa độ xy và uv như sau:

$$\begin{aligned} u &= kx + c_u \\ v &= ky + c_v \end{aligned} \quad (2.2)$$

Các thông số f, k, c_u, c_v được suy ra từ quá trình hiệu chỉnh thông số nội camera.

Quay lại bài toán giải tam giác. Xét hai ảnh stereo cùng chụp điểm đặc trưng P . Mỗi liên hệ giữa hai ảnh này được gọi là hình học epipolar như Hình 2.5. Hình chiếu điểm P tương ứng trên hai ảnh là $M_1 = [x_1 \ y_1]^T$ và $M_2 = [x_2 \ y_2]^T$. Từ mô hình pinhole camera (2.1), ta có vị trí điểm P trong hệ tọa độ camera 1 và 2 như sau:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = s_1 \begin{bmatrix} x_1 \\ y_1 \\ f_1 \end{bmatrix}, \quad \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = s_2 \begin{bmatrix} x_2 \\ y_2 \\ f_2 \end{bmatrix} \quad (2.3)$$

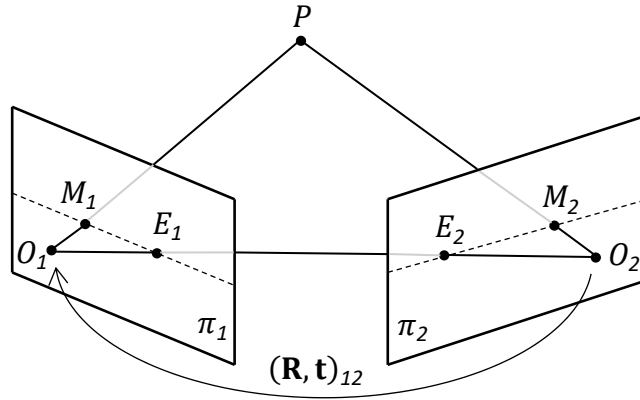
Quá trình hiệu chỉnh thông số ngoại camera cho biết ma trận xoay và vector tịnh tiến giữa hai camera $(\mathbf{R}, \mathbf{t})_{12}$, do đó:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \mathbf{R}_{12} \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} + \mathbf{t}_{12} \quad (2.4)$$

Kết hợp (2.3) và (2.4), ta giải hệ phương trình để tìm s_1, s_2 :

$$s_1 \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} - s_2 \left(\mathbf{R}_{12} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \right) = \mathbf{t}_{12} \quad (2.5)$$

Thay nghiệm s_1, s_2 vào (2.3), ta tái tạo được tọa độ 3D của điểm P .



Hình 2.5. Hình học epipolar

2.3. Thuật toán tối ưu hóa phi tuyến Powell's dogleg

Thuật toán tối ưu hóa phi tuyến – *nonlinear optimization* giải quyết bài toán ước lượng thông số của hệ thống phi tuyến sao cho hàm chi phí bình phương đạt giá trị

nhỏ nhất (nonlinear least square). Quy trình chung của thuật toán là dùng vòng lặp để thay đổi dần thông số theo hướng giảm của hàm chi phí. Trong đó, chiến lược chọn *bước*, hay lượng thay đổi của vector thông số trong mỗi vòng lặp, sẽ quyết định khả năng tìm nghiệm của thuật toán.

Gọi $\mathbf{x} \in \mathbb{R}^n$ là vector thông số, $\mathbf{e}(\mathbf{x}) = [e_1(\mathbf{x}) \dots e_m(\mathbf{x})]$ là vector hàm sai số dự báo. Bài toán đặt ra là tìm nghiệm tối ưu $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \{V(\mathbf{x})\}$ sao cho cực tiểu hàm mục tiêu:

$$V(\mathbf{x}) = \frac{1}{2} \|\mathbf{e}(\mathbf{x})\|^2 = \frac{1}{2} \sum_{i=1}^m e_i(\mathbf{x})^2 \quad (2.6)$$

Có nhiều thuật toán khả thi để giải bài toán này như suy giảm gradient, Gauss-Newton, Levenberg-Marquardt, Powell's dogleg, v.v.. Điểm chung của các phương pháp là thực hiện xấp xỉ tuyến tính hàm $V(\mathbf{x})$ trong vùng lân cận. Tuy nhiên, mỗi phương pháp có cách tính bước $\Delta\mathbf{x}$ khác nhau. Phương pháp suy giảm gradient bắt buộc $\Delta\mathbf{x}$ đi theo hướng mà $V(\mathbf{x})$ giảm sâu nhất. Trong khi đó, phương pháp Gauss-Newton xác định $\Delta\mathbf{x}$ sao cho mô hình tuyến tính của $V(\mathbf{x})$ đạt cực tiểu. Từ hai cách tiếp cận trên, các thuật toán miền tin cậy như Levenberg-Marquardt, Powell's dogleg tổng hợp $\Delta\mathbf{x}$ sao cho độ lớn của nó không vượt quá bán kính miền tin cậy μ , đảm bảo khả năng hội tụ của thuật toán. Đồng thời, miền tin cậy được cập nhật theo khả năng kéo giảm hàm V của mô hình tuyến tính qua mỗi vòng lặp. So sánh với thuật toán Levenberg-Marquardt, thuật toán Powell's dogleg cho thấy tốc độ hội tụ cao hơn [20]. Do vậy, để tìm nghiệm tối ưu một cách nhanh chóng và ổn định, đề tài áp dụng thuật toán tối ưu hóa Powell's dogleg được trình bày dưới đây.

Đầu tiên, ta khai triển Taylor hàm $\mathbf{e}(\mathbf{x})$ trong miền lân cận:

$$\mathbf{e}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{e}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\Delta\mathbf{x} \triangleq \mathbf{I}(\Delta\mathbf{x}) \quad (2.7)$$

với $\mathbf{J}(\mathbf{x})$ là ma trận Jacobian của $\mathbf{e}$ theo $\mathbf{x}$; $\mathbf{I}(\Delta\mathbf{x})$ là mô hình tuyến tính của $\mathbf{e}$ xung quanh $\mathbf{x}$. Gọi $L(\Delta\mathbf{x})$ là mô hình tuyến tính hóa của $V(\mathbf{x})$, thay vào (2.6):

$$V(\mathbf{x} + \Delta\mathbf{x}) \approx L(\Delta\mathbf{x}) \triangleq \frac{1}{2} \mathbf{I}(\Delta\mathbf{x})^\top \mathbf{I}(\Delta\mathbf{x}) \quad (2.8)$$

$$= V(\mathbf{x}) + \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{e} + \frac{1}{2} \Delta \mathbf{x}^T \mathbf{J}^T \mathbf{J} \Delta \mathbf{x}$$

Tiếp cận vấn đề theo phương pháp Gauss-Newton, đạo hàm bậc 1 và bậc 2 của L theo $\Delta \mathbf{x}$ là:

$$\begin{aligned} \mathbf{L}'(\Delta \mathbf{x}) &= \frac{\partial L}{\partial \Delta \mathbf{x}} = \mathbf{J}^T \mathbf{e} + \mathbf{J}^T \mathbf{J} \Delta \mathbf{x} \\ \mathbf{L}''(\Delta \mathbf{x}) &= \mathbf{J}^T \mathbf{J} \end{aligned} \quad (2.9)$$

Ta thấy nếu $\mathbf{J}$ có hạng đủ thì $\mathbf{L}''$ xác định dương và không phụ thuộc vào $\Delta \mathbf{x}$. Khi đó, hàm L có tiêu duy nhất là nghiệm của phương trình $\mathbf{L}' = 0$:

$$\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}) \Delta \mathbf{x} = -\mathbf{J}(\mathbf{x})^T \mathbf{e}(\mathbf{x}) \quad (2.10)$$

Nghiệm của phương trình trên được gọi là là *bước Gauss-Newton*:

$$\Delta \mathbf{x}_{\text{gn}} = -(\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x}))^{-1} \mathbf{J}(\mathbf{x})^T \mathbf{e}(\mathbf{x}) \quad (2.11)$$

Tiếp cận vấn đề theo phương pháp suy giảm gradient, hướng giảm dốc nhất của hàm V là chiều âm của gradient $\mathbf{g}(\mathbf{x}) = \mathbf{V}'(\mathbf{x})$:

$$\mathbf{h}_{\text{sd}} = -\mathbf{g}(\mathbf{x}) = -\mathbf{J}(\mathbf{x})^T \mathbf{e} \quad (2.12)$$

Gọi α là tỉ lệ độ dài của bước theo hướng $\mathbf{h}_{\text{sd}}$, xét mô hình tuyến tính L theo α :

$$L(\alpha \mathbf{h}_{\text{sd}}) = V(\mathbf{x}) + \alpha \mathbf{h}_{\text{sd}}^T \mathbf{J}^T \mathbf{e} + \frac{1}{2} \alpha^2 \|\mathbf{J}(\mathbf{x}) \mathbf{h}_{\text{sd}}\|^2 \quad (2.13)$$

Hàm này đạt cực tiểu khi $\mathbf{L}'(\alpha) = 0$, suy ra *bước giảm dốc nhất* theo phương pháp suy giảm gradient là:

$$\alpha = -\frac{\mathbf{h}_{\text{sd}}^T \mathbf{J}(\mathbf{x})^T \mathbf{e}(\mathbf{x})}{\|\mathbf{J}(\mathbf{x}) \mathbf{h}_{\text{sd}}\|^2} = \frac{\|\mathbf{g}(\mathbf{x})\|^2}{\|\mathbf{J}(\mathbf{x}) \mathbf{g}(\mathbf{x})\|^2} \quad (2.14)$$

$$\Delta \mathbf{x}_{\text{sd}} = \alpha \mathbf{h}_{\text{sd}} = -\frac{\|\mathbf{g}(\mathbf{x})\|^2}{\|\mathbf{J}(\mathbf{x}) \mathbf{g}(\mathbf{x})\|^2} \mathbf{g}(\mathbf{x}) \quad (2.15)$$

Từ hai bước $\Delta \mathbf{x}_{\text{gn}}$ và $\Delta \mathbf{x}_{\text{sd}}$ tìm được ở trên, phương pháp Powell's dogleg đề xuất một chiến lược chọn bước $\Delta \mathbf{x}_{\text{dl}}$ theo bán kính miền tin cậy μ như sau:

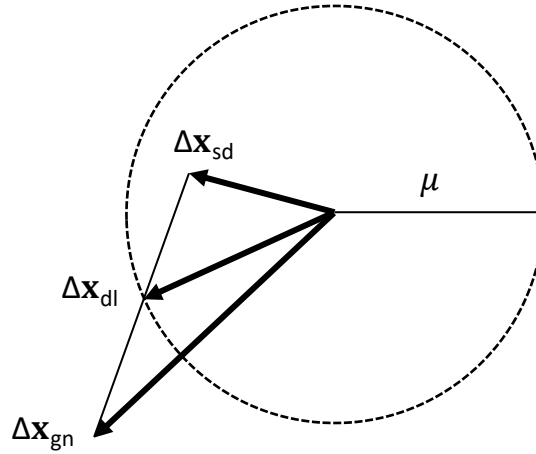
- Nếu $\|\Delta \mathbf{x}_{gn}\| \leq \mu$, chọn $\Delta \mathbf{x}_{dl} = \Delta \mathbf{x}_{gn}$: Khi miền tin cậy lớn hơn bước Gauss-Newton, mô hình xấp xỉ tuyến tính là đáng tin cậy, ta đi theo bước Gauss-Newton nhằm đạt tốc độ hội tụ gần như bình phương.
- Nếu $\|\Delta \mathbf{x}_{gn}\| > \mu$ và $\|\Delta \mathbf{x}_{sd}\| \geq \mu$, chọn $\Delta \mathbf{x}_{dl} = (\mu/\|\Delta \mathbf{x}_{sd}\|)\Delta \mathbf{x}_{sd}$: Khi bước Gauss-Newton không tin cậy, mà bước giảm dốc nhất của hàm chi phí thật vượt hơn miền tin cậy, ta đi theo hướng giảm dốc nhất với phạm vi μ để hàm giảm nhanh nhất.
- Các trường hợp còn lại, chọn $\Delta \mathbf{x}_{dl} = \Delta \mathbf{x}_{sd} + \beta(\Delta \mathbf{x}_{gn} - \Delta \mathbf{x}_{sd})$ sao cho $\|\Delta \mathbf{x}_{dl}\| = \mu$. Ta đi theo bước tổng hợp giữa bước Gauss-Newton và bước giảm dốc nhất, nhằm đạt tốc độ hội tụ cao nhất trong miền tin cậy, như Hình 2.6. Đặt $\mathbf{a} = \Delta \mathbf{x}_{sd}$, $\mathbf{b} = \Delta \mathbf{x}_{gn}$, $c = \mathbf{a}^T(\mathbf{b} - \mathbf{a})$, ta tính được β :

$$\beta = \begin{cases} \frac{-c + \sqrt{c^2 + \|\mathbf{b} - \mathbf{a}\|^2(\mu^2 - \|\mathbf{a}\|^2)}}{\|\mathbf{b} - \mathbf{a}\|^2} & (c \leq 0) \\ \frac{\mu^2 - \|\mathbf{a}\|^2}{c + \sqrt{c^2 + \|\mathbf{b} - \mathbf{a}\|^2(\mu^2 - \|\mathbf{a}\|^2)}} & (c > 0) \end{cases} \quad (2.16)$$

Sau mỗi vòng lặp, miền tin cậy được điều chỉnh theo tiêu chuẩn ϱ :

$$\varrho = \frac{V(\mathbf{x}) - V(\mathbf{x} + \Delta \mathbf{x}_{dl})}{L(\mathbf{0}) - L(\Delta \mathbf{x}_{dl})} \quad (2.17)$$

Nếu ϱ lớn, hàm $V(\mathbf{x} + \Delta \mathbf{x}_{dl})$ giảm tốt với mô hình tuyến tính $L(\Delta \mathbf{x}_{dl})$, ta có thể nói rộng miền tin cậy. Ngược lại, khi ϱ nhỏ hoặc âm cho thấy $V(\mathbf{x} + \Delta \mathbf{x}_{dl})$ giảm kém, ta cần thu hẹp miền tin cậy.



Hình 2.6. Bước thay đổi tổng hợp trong thuật toán Powell's dogleg

Thuật toán kết thúc khi thỏa một trong những điều kiện sau: (1) gradient hoặc (2) sai số đã giảm nhỏ hơn ngưỡng cần thiết, (3) bước thay đổi giữa hai lần lặp không còn đáng kể, hoặc (4) vượt quá số lần lặp tối đa.

$$\begin{aligned}
 &1. \|g\|_{\infty} \leq \varepsilon_1 \\
 &2. \|\Delta x_{dl}\| \leq \varepsilon_2(\|x\| + \varepsilon_2) \\
 &3. \|e(x)\|_{\infty} \leq \varepsilon_3 \\
 &4. k > k_{max}
 \end{aligned} \tag{2.18}$$

Dưới đây là bảng tóm tắt thuật toán.

Bảng 2.1. Tóm tắt thuật toán Powell's dogleg

Init $x_0, \mu_0, \varepsilon_1, \varepsilon_2, \varepsilon_3, k_{max}$ $k = 0;$ $x = x_0;$ $\mu = \mu_0;$ $g = J(x)^T e(x)$ $found = (\ e(x)\ _{\infty} \leq \varepsilon_3) \text{ or } (\ g\ _{\infty} \leq \varepsilon_1)$ while (not found) and ($k < k_{max}$) do $k = k + 1;$ $\Delta x_{gn} = -(J(x)^T J(x))^{-1} J(x)^T e(x)$ $\Delta x_{sd} = -(\ g(x)\ ^2 g(x)) / \ J(x)g(x)\ ^2$ if $\ \Delta x_{gn}\ \leq \mu$ then $\Delta x_{dl} = \Delta x_{gn}$ elseif $\ \Delta x_{sd}\ \geq \mu$ then $\Delta x_{dl} = (\mu / \ \Delta x_{sd}\) \Delta x_{sd}$ else $\Delta x_{dl} = \Delta x_{sd} + \beta(\Delta x_{gn} - \Delta x_{sd})$ endif if $\ \Delta x_{dl}\ \leq \varepsilon_2(\ x\ + \varepsilon_2)$ then
--

```
        found = true
    else
         $\mathbf{x}_{\text{new}} = \mathbf{x} + \Delta \mathbf{x}_{\text{dl}}$ 
         $\varrho = (V(\mathbf{x}) - V(\mathbf{x}_{\text{new}})) / (L(\mathbf{0}) - L(\Delta \mathbf{x}_{\text{dl}}))$ 
        if  $\varrho > 0$  then
             $\mathbf{x} = \mathbf{x}_{\text{new}}; \quad \mathbf{g} = \mathbf{J}(\mathbf{x})^T \mathbf{e}(\mathbf{x})$ 
            found =  $(\|\mathbf{e}(\mathbf{x})\|_{\infty} \leq \varepsilon_3)$  or  $(\|\mathbf{g}\|_{\infty} \leq \varepsilon_1)$ 
        endif
        if  $\varrho > 0.75$  then
             $\mu = \max\{\mu, 3 * \|\Delta \mathbf{x}_{\text{dl}}\|\}$ 
        elseif  $\varrho < 0.25$  then
             $\mu = 0.5 * \mu;$ 
            found =  $(\|\mathbf{e}(\mathbf{x})\|_{\infty} \leq \varepsilon_3)$  or  $(\|\mathbf{g}\|_{\infty} \leq \varepsilon_1)$ 
        endif
    endif
endwhile
Return  $\mathbf{x}$ 
```

CHƯƠNG 3. HỆ THỐNG ĐỊNH VỊ TÍCH HỢP THỊ GIÁC LẬP THỂ VÀ QUÁN TÍNH

3.1. Hiệu chuẩn IMU/StereoCamera

3.1.1. Đồng bộ thời gian IMU/Camera

Để xây dựng hệ thống tích hợp chặt nói chung, việc đồng bộ thời gian là rất quan trọng. Nếu các cảm biến không được đồng bộ chính xác, sai số thời gian sẽ chuyển thành sai số trong không gian và làm suy giảm hệ tích hợp.

Độ lệch thời gian giữa IMU và Camera được gây ra bởi ba yếu tố chính: thời gian màn trập camera, thời gian xử lý nội IMU và các độ trễ giao tiếp. Trong đó, hai yếu tố sau thường là hằng số và tổng của chúng có thể được ước lượng bằng các thuật toán hiệu chuẩn thời gian. Ngược lại, thời gian màn trập camera thay đổi theo điều kiện sáng nên cần được đọc liên tục nhằm điều chỉnh thời điểm kích đồng bộ camera. Nếu đặt thời gian màn trập cố định, camera sẽ không thích nghi được khi cường độ sáng trong môi trường thay đổi, dẫn đến chất lượng ảnh bị suy giảm.

Để thực hiện đồng bộ IMU/Camera, trước tiên ta cho IMU phát ra xung clock đồng bộ với thời điểm đọc dữ liệu quán tính. Xung clock sau đó đi qua mạch xử lý chia tần số xuống bằng tốc độ khung hình của camera. Thời gian màn trập hiện tại của camera cũng được đọc vào mạch xử lý, kết hợp với các độ trễ cố định để tính ra thời gian cần dịch trên xung clock đã chia tần, tạo ra xung kích phù hợp cho camera.

Cụ thể, gọi t_r là thời điểm dữ liệu IMU, t_p là thời điểm dữ liệu camera và t_q là thời điểm xung kích. Mặt khác, gọi I là gốc thời gian của việc đọc dữ liệu IMU, C là gốc thời gian của việc đọc dữ liệu camera, và W là gốc thời gian đúng. Các độ trễ trong hệ IMU/Camera có công thức như sau:

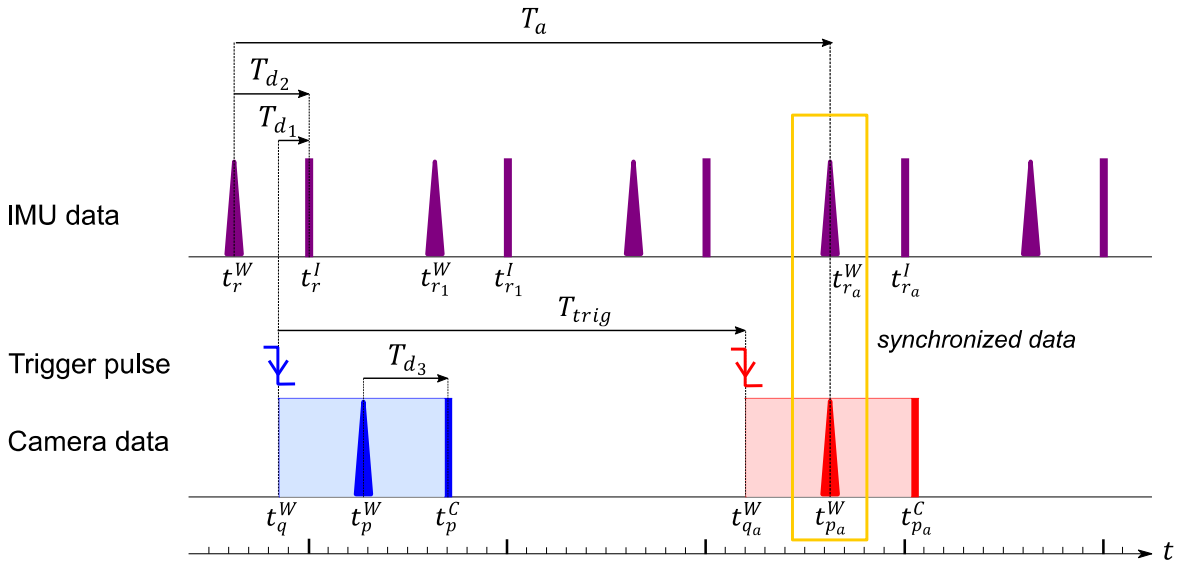
$$T_{d_1} = t_r^I - t_q^W \quad (3.1)$$

$$T_{d_2} = t^I - t^W = t_r^I - t_r^W \quad (3.2)$$

$$T_{d_3} = t^C - t^W = t_p^C - t_p^W = \frac{T_{shutter}}{2} \quad (3.3)$$

Trong đó, T_{d_1} thể hiện độ lệch trong thời gian truyền dữ liệu IMU và dữ liệu camera đến máy tính, được tính bằng thời điểm nhận dữ liệu IMU trừ đi thời điểm xung kích (cũng là thời điểm camera bắt đầu chụp ảnh). T_{d_2} thể hiện thời gian xử lý nội IMU so với thời điểm thực của dữ liệu quán tính. T_{d_3} thể hiện thời gian trễ của camera bằng $\frac{1}{2}$ thời gian màn trập. Giả sử vận tốc của camera so với môi trường là hằng số trong thời gian màn trập rất ngắn (vài ms), việc lấy thời điểm $\frac{1}{2}$ thời gian màn trập sẽ đại diện tốt nhất cho chuyển động trong ảnh.

Các độ trễ và thời gian bù được minh họa trong Hình 3.1. Trong đó, các thanh chữ nhật đánh dấu thời điểm nhận dữ liệu, các thanh tam giác đánh dấu thời điểm đúng của dữ liệu. Xung kích được thể hiện bởi dấu cạnh xuống. Màu xanh dương đại diện cho các thời điểm camera ban đầu, khi xung kích chưa được hiệu chỉnh thời gian. Màu đỏ đại diện cho các thời điểm camera khi xung kích đã được hiệu chỉnh. Màu tím đại diện cho các thời điểm IMU. Khung chữ nhật vàng thể hiện một cặp dữ liệu IMU-Camera đã được đồng bộ đúng.



Hình 3.1. Giản đồ đồng bộ thời gian IMU/Camera

Để tính các độ trễ trong hệ, ta sẽ tìm các khoảng cách thời gian sau:

$$d_L = t^I - t^C = \frac{-T_{shutter}}{2} + T_{d_2} \quad (3.4)$$

$$d_S = t_r^I - t_p^C = -T_{shutter} + T_{d_1} \quad (3.5)$$

với d_L là khoảng cách giữa dòng thời gian thực IMU và camera, d_S là khoảng cách giữa thời điểm nhận dữ liệu IMU và camera. Để tính các độ trễ một cách chính xác, ta đặt thời gian màn trập của camera là cố định. Khi đó, d_S được đo bằng hiệu giữa hai thời điểm nhận dữ liệu IMU và camera tương ứng. Từ đó ta suy ra T_{d_1} nhờ biết trước $T_{shutter}$. Mặt khác, d_L được ước lượng bằng thuật toán hiệu chuẩn thời gian trong thư viện Kalibr [13]. Đây là bộ công cụ có nhiều cải tiến trong việc hiệu chuẩn IMU/Camera. Có d_L ta tìm được T_{d_2} .

Từ các giá trị trên, ta có thể căn chỉnh thời điểm IMU và camera bằng cách dịch xung kích đi một khoảng thích hợp T_{trig} :

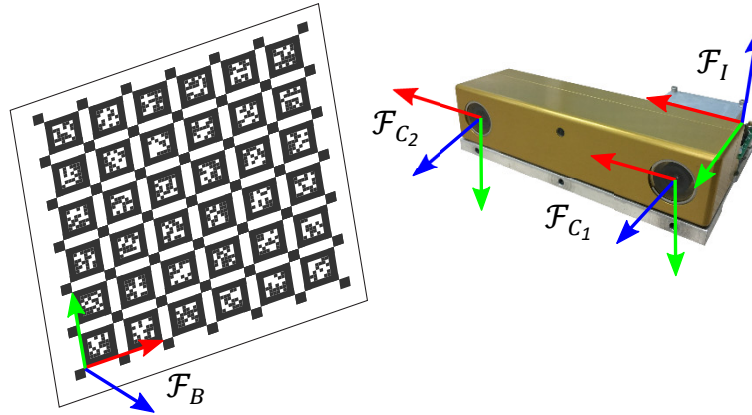
$$T_{trig} = t_{q_a}^W - t_q^W = T_a + T_{d_1} - T_{d_2} - \frac{T_{shutter}}{2} = T_a - d_L + d_S \quad (3.6)$$

Trong đó $t_{q_a}^W$ là thời điểm xung kích sau khi dịch. Thay vì căn chỉnh thời điểm camera trùng với thời điểm IMU đầu tiên t_r^W , ta sẽ chọn một mẫu IMU sau đó $t_{r_a}^W$ sao cho T_{trig} luôn dương. Việc chọn mẫu này nhằm đơn giản hóa khâu thực hiện, trong khi không tạo thêm độ trễ trong hệ thống. Thời gian offset T_a được chọn là bội số của chu kì lấy mẫu IMU nhưng không lớn hơn chu kì lấy mẫu camera.

3.1.2. Hiệu chuẩn không gian IMU/StereoCamera

Thuật toán định vị thị giác lập thể-quán tính cần biết trước các thông số không gian của stereo camera cũng như liên hệ giữa stereo camera và IMU để thực hiện ước lượng. Các thông số này được tìm thông qua quá trình hiệu chuẩn. Hình 3.2 mô tả bố trí để hiệu chuẩn hệ thống IMU/StereoCamera. Trong đó, $\mathcal{F}_A$ là hệ tọa độ của mẫu hiệu chuẩn (calibration pattern), trong trường hợp này là AprilTag [14]. $\mathcal{F}_{C_1}$ và $\mathcal{F}_{C_2}$ lần lượt là hệ tọa độ hai camera, và $\mathcal{F}_I$ là hệ tọa độ IMU. Stereo camera và IMU được gắn chặt với nhau để tư thế giữa chúng là không thay đổi. Việc hiệu chuẩn được chia

thành hai bước gồm camera-to-camera: tìm thông số nội từng camera và tư thế giữa hai camera, và camera-to-IMU: tìm tư thế giữa camera và IMU.



Hình 3.2. Các hệ tọa độ của hệ thống IMU/StereoCamera và mẫu hiệu chuẩn

Trong bước camera-to-camera, ta thu thập các cặp ảnh chụp mẫu hiệu chỉnh di chuyển trước camera đứng yên. Để có kết quả tốt, ta cần di chuyển mẫu hiệu chỉnh đến nhiều vị trí và hướng xoay khác nhau, đồng thời đi qua toàn bộ diện tích trên ảnh. Thư viện Kalibr [13] được sử dụng để tính toán thông số hiệu chuẩn, bao gồm mô hình chiếu của mỗi camera $\mathbf{h}_1(\cdot)$, $\mathbf{h}_2(\cdot)$ dựa trên mô hình pinhole và méo dạng cách đều (equidistant), và tư thế stereo $\mathbf{T}_{C_1C_2}$.

Để hiệu chuẩn camera-to-IMU, ta thu thập dữ liệu camera và IMU khi di chuyển dùng đưa camera trước vật mẫu cố định. Chuyển động cần mượt mà và bao phủ hết các trục tịnh tiến cũng như góc xoay của IMU với vận tốc vừa đủ để đảm bảo các thông số được quan sát tốt. Bên cạnh đó, cần duy trì điều kiện ánh sáng tốt và thời gian màn trập nhỏ để hạn chế độ nhòe trên ảnh. Để cấu hình thông số nhiễu IMU cho việc tính toán hiệu chuẩn, ta thực hiện thí nghiệm đặt IMU đứng yên ở nhiều tư thế, mỗi tư thế trong 5 phút để tìm giá trị nhiễu thực tế. Sau cùng, thư viện Kalibr ước lượng tư thế camera-IMU bằng phương pháp tối ưu hóa chung, cùng với tư thế IMU theo thời gian, độ lệch (bias) của cảm biến gia tốc (accelerometer) và cảm biến vận tốc góc (gyroscope). Trong đó tư thế, gia tốc và vận tốc góc IMU được tham số hóa theo đường cong B-spline, trong khi các bias được mô hình là thay đổi ngẫu nhiên (random walk).

Kết quả hiệu chuẩn camera-to-camera:

- Thông số mô hình chiếu và thông số méo dạng tìm được của mỗi camera:

	fx [pixel]	fy [pixel]	cu [pixel]	cv [pixel]
Cam1	528.178 ± 0.204	528.097 ± 0.202	321.421 ± 0.295	251.251 ± 0.437
Cam2	530.358 ± 0.204	530.216 ± 0.203	322.143 ± 0.301	252.185 ± 0.424

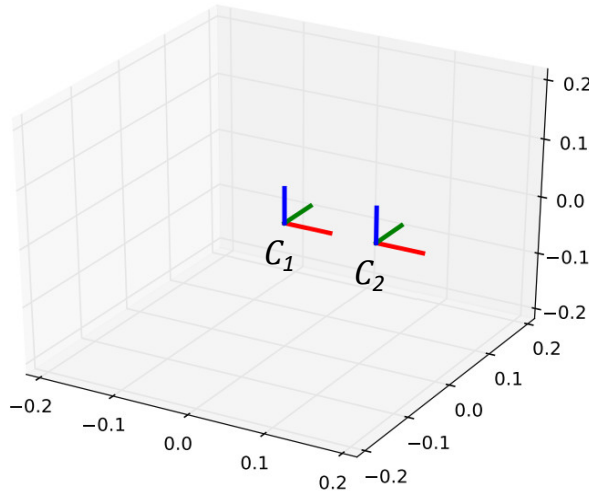
Bảng 3.1. Thông số mô hình chiếu pinhole camera

	k1	k2	k3	k4
Cam1	-0.0490 ± 0.0103	0.0737 ± 0.0733	-0.1319 ± 0.2043	0.0794 ± 0.1947
Cam2	-0.0317 ± 0.0100	-0.0593 ± 0.0723	0.2170 ± 0.2055	-0.2277 ± 0.2004

Bảng 3.2. Thông số méo dạng equidistant camera

- Tư thế stereo:

$$T_{C_1C_2} = \begin{bmatrix} 1.0000 & 0.0007 & -0.0028 & 0.1198 \\ -0.0007 & 1.0000 & 0.0015 & -0.0005 \\ 0.0028 & -0.0015 & 1.0000 & 0.0008 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Hình 3.3. Hệ stereo camera tìm được từ quá trình hiệu chuẩn

- Sai số reprojection (RMS): Cam1: 0.1392 pixel, Cam2: 0.1339 pixel.

Nhận xét: Kết quả hiệu chuẩn phù hợp với thực tế. Baseline giữa hai camera tìm được là 0.1198m, rất gần với giá trị theo datasheet là 12cm. Các sai số mô hình chiếu, méo dạng cũng như sai số reprojection là rất nhỏ, chứng tỏ quá trình hiệu chuẩn đạt kết quả tốt.

Kết quả hiệu chuẩn camera-to-IMU:

- Tư thế IMU-cam1 và IMU-cam2:

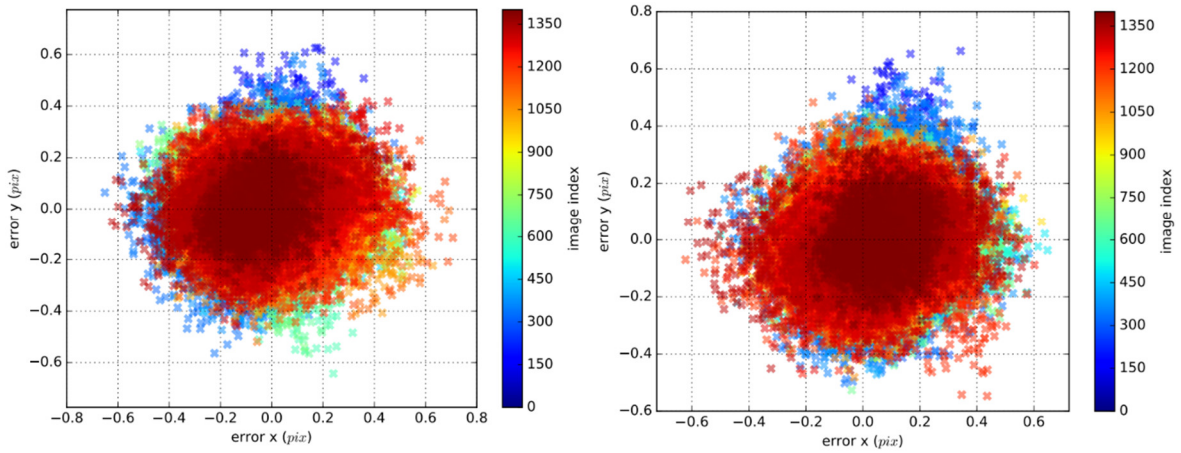
$$T_{IC_1} = \begin{bmatrix} 0.9999 & -0.0094 & -0.0102 & -0.0208 \\ 0.0102 & -0.0004 & 0.9999 & 0.0517 \\ -0.0094 & -1.0000 & -0.0003 & -0.0050 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{IC_2} = \begin{bmatrix} 0.9999 & -0.0086 & -0.0131 & 0.0991 \\ 0.0131 & -0.0019 & 0.9999 & 0.0538 \\ -0.0087 & -1.0000 & -0.0017 & -0.0057 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Sai số hiệu chuẩn (RMS):

Reprojection error [pixel]		Gyroscope [rad/s]	Accelerometer [m/s ²]
Cam1	Cam2		
0.1442	0.1376	0.000488	0.009173

Bảng 3.3. Sai số hiệu chuẩn camera-to-IMU



Hình 3.4. Phân bố sai số reprojection trên camera trái và phải trong bước hiệu chuẩn camera-to-IMU

Các sai số reprojection, gyroscope và accelerometer đều rất nhỏ cho thấy quá trình hiệu chuẩn camera-to-IMU đạt kết quả tốt. Như vậy, hệ cảm biến IMU/StereoCamera đã được hiệu chỉnh thành công, sẵn sàng cho việc tích hợp.

3.2. Thuật toán tích hợp IMU/Camera

3.2.1. Trạng thái hệ tích hợp

Trạng thái ước lượng trong hệ tích hợp bao gồm trạng thái robot $\mathbf{x}_R$ và vị trí các landmark $\mathbf{x}_L^j$:

$$\mathbf{x}_R = [\mathbf{p}_{WI}^T \quad \mathbf{q}_{WI}^T \quad {}_I\mathbf{v}^T \quad \mathbf{b}_\omega^T \quad \mathbf{b}_a^T]^T \in \mathbb{R}^3 \times SO(3) \times \mathbb{R}^9 \quad (3.7)$$

$$\mathbf{x}_L^j = {}_W\mathbf{l}^j \in \mathbb{R}^4 \quad (3.8)$$

trong đó $\mathbf{p}_{WI}$ và $\mathbf{q}_{WI}$ là vị trí và hướng của IMU $\mathcal{F}_I$ trong hệ tọa độ thế giới $\mathcal{F}_W$, ${}_I\mathbf{v}$ là vận tốc IMU so với chính nó, $\mathbf{b}_\omega$ và $\mathbf{b}_a$ là mức bias của cảm biến vận tốc góc và gia tốc, và ${}_W\mathbf{l}^j$ là vị trí thuần nhất của landmark thứ j so với mặt đất.

Trạng thái thông số ngoại của camera thứ i là $\mathbf{x}_{C_i}$, thu được từ quá trình hiệu chuẩn không gian IMU/Camera hoặc ước lượng online theo thời gian:

$$\mathbf{x}_{C_i} = [\mathbf{p}_{IC_i}^T \quad \mathbf{q}_{IC_i}^T]^T \in \mathbb{R}^3 \times SO(3) \quad (3.9)$$

Để tiện biểu diễn, ta đặt hai thành phần con của vector trạng thái là tư thế robot $\mathbf{x}_T$ và vận tốc-bias IMU $\mathbf{x}_{sb}$ là:

$$\mathbf{x}_T = [\mathbf{p}_{WI}^T \quad \mathbf{q}_{WI}^T]^T \quad (3.10)$$

$$\mathbf{x}_{sb} = [{}_I\mathbf{v}^T \quad \mathbf{b}_\omega^T \quad \mathbf{b}_a^T]^T \quad (3.11)$$

Lượng thay đổi (perturbation) của trạng thái quanh điểm làm việc $\bar{\mathbf{x}}$ được định nghĩa là $\delta\mathbf{x} = \mathbf{x} - \bar{\mathbf{x}}$. Để biểu diễn trạng thái với số biến tối thiểu, ta đưa ra các ngoại lệ là: sai số hướng xoay robot được tính bằng chênh lệch theo vector trục-góc (axis-angle) $\delta\boldsymbol{\alpha}$, và sai số landmark chỉ lấy ba thành phần đầu khác 0.

$$\delta\mathbf{x}_R = [\delta\mathbf{p}_{WI}^T \quad \delta\boldsymbol{\alpha}_{WI}^T \quad \delta{}_I\mathbf{v}^T \quad \delta\mathbf{b}_\omega^T \quad \delta\mathbf{b}_a^T]^T \in \mathbb{R}^{15} \quad (3.12)$$

$$\delta\mathbf{x}_L^j = {}_W\delta\mathbf{l}_{1:3}^j \in \mathbb{R}^3 \quad (3.13)$$

3.2.2. Mô hình động học IMU

Mô hình động học IMU được dùng để lan truyền trạng thái ước lượng ở tần số cao. Ta có trạng thái IMU cũng là trạng thái robot tại (3.7):

$$\mathbf{x}_R = [\mathbf{p}_{WI}^T \quad \mathbf{q}_{WI}^T \quad {}_I\mathbf{v}^T \quad \mathbf{b}_\omega^T \quad \mathbf{b}_a^T]^T \quad (3.14)$$

Đặt ${}_I\boldsymbol{\omega}$ là vận tốc góc và ${}_I\mathbf{a}$ là gia tốc của IMU trên 3 trục với các giá trị đo tương ứng ${}_I\tilde{\boldsymbol{\omega}}$ và ${}_I\tilde{\mathbf{a}}$, ${}_W\mathbf{g}$ là vector gia tốc trọng trường, $\mathbf{w}_\omega$ và $\mathbf{w}_a$ là nhiễu trắng Gaussian trên phép đo vận tốc góc và gia tốc, $\mathbf{w}_{b_\omega}$ và $\mathbf{w}_{b_a}$ là nhiễu trắng trên lượng thay đổi ngẫu nhiên trên bias của phép đo vận tốc góc và gia tốc. Mô hình động học phi tuyến của IMU được cho bởi:

$$\begin{cases} \dot{\mathbf{p}}_{WI} = \mathbf{R}_{WI} \mathbf{v} \\ \dot{\mathbf{q}}_{WI} = \boldsymbol{\Omega}(\boldsymbol{\omega}^j) \mathbf{q}_{WI} \\ {}_I\dot{\mathbf{v}} = {}_I\mathbf{a} + \mathbf{R}_{WI} {}_W\mathbf{g} - [{}_I\boldsymbol{\omega}]^\times {}_I\mathbf{v} \\ \dot{\mathbf{b}}_\omega = \mathbf{w}_{b_\omega} \\ \dot{\mathbf{b}}_a = \mathbf{w}_{b_a} \end{cases} \quad (3.15)$$

Trong đó gia tốc và vận tốc góc thực tế là:

$$\begin{aligned} {}_I\mathbf{a} &= {}_I\tilde{\mathbf{a}} - \mathbf{b}_a + \mathbf{w}_a \\ {}_I\boldsymbol{\omega} &= {}_I\tilde{\boldsymbol{\omega}} - \mathbf{b}_\omega + \mathbf{w}_\omega \end{aligned} \quad (3.16)$$

Ta có ma trận $\boldsymbol{\Omega}(\boldsymbol{\omega})$:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} [\boldsymbol{\omega}]^\times & -\boldsymbol{\omega} \\ \boldsymbol{\omega}^T & 0 \end{bmatrix} \quad (3.17)$$

Tuyến tính hóa phương trình phi tuyến IMU quanh điểm làm việc $\bar{\mathbf{x}}_R$, ta thu được phương trình trạng thái xấp xỉ tuyến tính trong miền thời gian liên tục của IMU:

$$\delta \dot{\mathbf{x}}_R \approx \mathbf{F}_c(\bar{\mathbf{x}}_R) \delta \mathbf{x}_R + \mathbf{G}_c(\bar{\mathbf{x}}_R) \mathbf{w} \quad (3.18)$$

với $\delta \mathbf{x}$ là lượng thay đổi (perturbation) của vector trạng thái quanh điểm làm việc. Các ma trận Jacobian $\mathbf{F}_c$ và $\mathbf{G}_c$ lần lượt là:

$$\mathbf{F}_c = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & [\bar{\mathbf{R}}_{WI} \bar{\mathbf{v}}]^\times & \bar{\mathbf{R}}_{WI} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & -\bar{\mathbf{R}}_{WI} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -\bar{\mathbf{R}}_{WI}^T [\mathbf{w} \mathbf{g}]^\times & -[\mathbf{I} \boldsymbol{\omega}]^\times & -[\mathbf{I} \mathbf{v}]^\times & -\mathbf{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (3.19)$$

$$\mathbf{G}_c = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \bar{\mathbf{R}}_{WI} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_3 \end{bmatrix} \quad (3.20)$$

Biểu thức $\mathbf{G}_c$ được suy ra đơn giản từ (3.15). Ta chứng minh $\mathbf{F}_c$ sử dụng một số tính chất của ma trận xoay và tích chéo trong [21] như sau:

$$\mathbf{F}_{12} = \frac{\partial (\dot{\mathbf{p}}_{WI})}{\partial \boldsymbol{\alpha}_{WI}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (\mathbf{R}_{WI} \mathbf{v})}{\partial \boldsymbol{\alpha}_{WI}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = [\mathbf{R}_{WI} \mathbf{v}]^\times \mathbf{S}(\boldsymbol{\alpha}_{WI}) \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = [\bar{\mathbf{R}}_{WI} \bar{\mathbf{v}}]^\times \quad (3.21)$$

$$\mathbf{F}_{13} = \frac{\partial (\dot{\mathbf{p}}_{WI})}{\partial \mathbf{I} \mathbf{v}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (\mathbf{R}_{WI} \mathbf{v})}{\partial \mathbf{I} \mathbf{v}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \bar{\mathbf{R}}_{WI} \quad (3.22)$$

$$\mathbf{F}_{24} = \frac{\partial (\dot{\boldsymbol{\alpha}}_{WI})}{\partial \mathbf{b}_g} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (\mathbf{R}_{WI} (\mathbf{I} \tilde{\boldsymbol{\omega}} + \mathbf{w}_g - \mathbf{b}_g))}{\partial \mathbf{b}_g} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = -\bar{\mathbf{R}}_{WI} \quad (3.23)$$

$$\begin{aligned} \mathbf{F}_{32} &= \frac{\partial (\mathbf{I} \mathbf{v})}{\partial \boldsymbol{\alpha}_{WI}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (\mathbf{R}_{WI} \mathbf{w}_g)}{\partial \boldsymbol{\alpha}_{WI}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \approx \frac{\partial (\bar{\mathbf{R}}_{WI}^T (1 + [\boldsymbol{\alpha}_{WI}]^\times) \mathbf{w}_g)}{\partial \boldsymbol{\alpha}_{WI}} \\ &= -\frac{\partial (\bar{\mathbf{R}}_{WI}^T [\mathbf{w}_g]^\times \boldsymbol{\alpha}_{WI})}{\partial \boldsymbol{\alpha}_{WI}} = -\bar{\mathbf{R}}_{WI}^T [\mathbf{w}_g]^\times \end{aligned} \quad (3.24)$$

$$\mathbf{F}_{33} = \frac{\partial (\mathbf{I} \dot{\mathbf{v}})}{\partial \mathbf{I} \mathbf{v}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (-[\mathbf{I} \boldsymbol{\omega}]^\times \mathbf{I} \mathbf{v})}{\partial \mathbf{I} \mathbf{v}} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = -[\mathbf{I} \bar{\boldsymbol{\omega}}]^\times \quad (3.25)$$

$$\mathbf{F}_{34} = \frac{\partial (\mathbf{I} \dot{\mathbf{v}})}{\partial \mathbf{b}_g} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (-[\mathbf{I} \boldsymbol{\omega}]^\times \mathbf{I} \mathbf{v})}{\partial \mathbf{b}_g} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial ([\mathbf{I} \mathbf{v}]^\times \mathbf{I} \boldsymbol{\omega})}{\partial \mathbf{b}_g} \Big|_{\mathbf{x}=\bar{\mathbf{x}}} \quad (3.26)$$

$$\begin{aligned}
 &= \frac{\partial \left([{}_I \mathbf{v}]^\times (\tilde{\boldsymbol{\omega}} + \mathbf{w}_g - \mathbf{b}_g) \right)}{\partial \mathbf{b}_g} \bigg|_{\mathbf{x}=\bar{\mathbf{x}}} = -[{}_I \bar{\mathbf{v}}]^\times \\
 \mathbf{F}_{35} &= \frac{\partial ({}_I \dot{\mathbf{v}})}{\partial \mathbf{b}_a} \bigg|_{\mathbf{x}=\bar{\mathbf{x}}} = \frac{\partial (\tilde{\mathbf{a}} - \mathbf{b}_a + \mathbf{w}_a + \mathbf{R}_{WI} \mathbf{w}_g - [{}_I \boldsymbol{\omega}]^\times {}_I \mathbf{v})}{\partial \mathbf{b}_a} \bigg|_{\mathbf{x}=\bar{\mathbf{x}}} = -\mathbf{I}_3
 \end{aligned} \tag{3.27}$$

3.2.3. Hàm mục tiêu tối ưu hóa

Tại mỗi thời điểm có giá trị đo IMU, trạng thái hệ thống được lan truyền theo phương trình động học (3.15). Sau đó, khi có phép đo camera, các trạng thái sẽ được tối ưu hóa theo hàm mục tiêu tổng hợp giữa sai số reprojection của camera $\mathbf{e}_{\text{cam}}$ và sai số IMU $\mathbf{e}_{\text{imu}}$:

$$J(\mathbf{x}) = \sum_{i=1}^I \sum_{k=1}^K \sum_{j \in \mathcal{J}(i,k)} \mathbf{e}_{\text{cam}}^{i,j,kT} \mathbf{W}_{\text{cam}}^{i,j,k} \mathbf{e}_{\text{cam}}^{i,j,k} + \sum_{k=1}^{K-1} \mathbf{e}_{\text{imu}}^kT \mathbf{W}_{\text{imu}}^k \mathbf{e}_{\text{imu}}^k \tag{3.28}$$

với i là chỉ số camera, k là chỉ số khung ảnh, j là chỉ số của landmark trong tập hợp các landmark $\mathcal{J}(i, k)$ trích được từ khung ảnh k chụp bởi camera i . $\mathbf{W}_{\text{rep}}$ và $\mathbf{W}_{\text{imu}}$ là các ma trận trọng số thể hiện mức độ tin cậy của phép đo tương ứng, tính bằng nghịch đảo của covariance nhiễu đo $\mathbf{W} = \mathbf{R}^{-1}$. Sai số IMU là sai lệch giữa trạng thái dự báo $\hat{\mathbf{x}}^{k+1}$ và trạng thái thực $\mathbf{x}^{k+1}$:

$$\mathbf{e}_{\text{imu}}^k = \hat{\mathbf{x}}_{\text{R}}^{k+1} - \mathbf{x}_{\text{R}}^{k+1} \tag{3.29}$$

Sai số reprojection được định nghĩa là chênh lệch vị trí 2D giữa các keypoint với các landmark 3D tương ứng được chiếu trở lại mặt phẳng ảnh:

$$\mathbf{e}_{\text{cam}}^{i,j,k} = \mathbf{z}^{i,j,k} - \mathbf{h}_i(\mathbf{T}_{C_iI}^k \mathbf{T}_{IW}^k \mathbf{l}^j) \tag{3.30}$$

Trong đó, $\mathbf{z}^{i,j,k}$ là vị trí 2D của keypoint trên mặt ảnh, ${}_W \mathbf{l}^j$ là vị trí 3D của landmark trong không gian. Các landmark đã tìm từ quá trình xử lý ảnh được chiếu trở lại ảnh 2D thông qua mô hình camera $\mathbf{h}_i(\cdot)$, có tính đến méo dạng phi tuyến của thấu kính. $\mathbf{T}_{C_iI}^k$ là phép biến đổi tọa độ camera/IMU tìm được từ quá trình hiệu chuẩn, và $\mathbf{T}_{IW}^k$ là nghịch đảo của tư thế robot cần ước lượng.

3.2.4. Hàm sai số camera

Để thực hiện tối ưu hóa, ta tìm các ma trận Jacobian của $\mathbf{e}_{\text{cam}}$ theo các chênh lệch trạng thái $\delta \mathbf{x}_T$, $\delta \mathbf{x}_L$ và $\delta \mathbf{x}_{C_i}$. Gọi $\mathbf{J}_{\text{cam},i}$ là ma trận Jacobian của mô hình đo camera $\mathbf{h}_i$ (.) đối với một landmark trong hệ tọa độ thuần nhất, ta có:

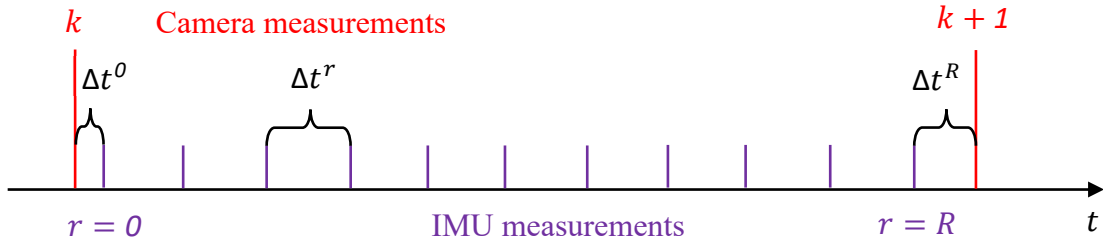
$$\frac{\partial \mathbf{e}_{\text{cam}}^{i,j,k}}{\partial \delta \mathbf{x}_T^k} = \mathbf{J}_{\text{cam},i} \bar{\mathbf{T}}_{C_i I}^k \begin{bmatrix} \bar{\mathbf{R}}_{IW}^k \bar{\mathbf{I}}_4^j & \bar{\mathbf{R}}_{IW}^k [{}_W \bar{\mathbf{I}}_{1:3}^j - \mathbf{p}_{WIW}^k \bar{\mathbf{I}}_4^j]^\times \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (3.31)$$

$$\frac{\partial \mathbf{e}_{\text{cam}}^{i,j,k}}{\partial \delta \mathbf{x}_L^j} = -\mathbf{J}_{\text{cam},i} \bar{\mathbf{T}}_{C_i I}^k \begin{bmatrix} \bar{\mathbf{R}}_{IW}^k \\ \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (3.32)$$

$$\frac{\partial \mathbf{e}_{\text{cam}}^{i,j,k}}{\partial \delta \mathbf{x}_{C_i}^k} = \mathbf{J}_{\text{cam},i} \begin{bmatrix} \bar{\mathbf{R}}_{C_i I}^k \bar{\mathbf{I}}_4^j & \bar{\mathbf{R}}_{C_i I}^k [{}_I \bar{\mathbf{I}}_{1:3}^j - \mathbf{p}_{IC_i I}^k \bar{\mathbf{I}}_4^j]^\times \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (3.33)$$

3.2.5. Hàm sai số IMU

Hình 3.5 mô tả giản đồ lấy mẫu của IMU và camera theo thời gian. Giữa hai phép đo camera từ k đến $k + 1$, ta đánh số các mẫu IMU là $r = 0, 1, \dots, R$. Gọi Δt^r là thời gian giữa hai mẫu IMU liên tiếp, Δt^0 là thời gian từ k đến mẫu IMU đầu tiên, và Δt^R là thời gian từ mẫu IMU cuối cùng đến $k + 1$.



Hình 3.5. Giản đồ lấy mẫu IMU và Camera

Đặt sai số IMU $\mathbf{e}_{\text{imu}}^k(\mathbf{x}_R^k, \mathbf{x}_R^{k+1}, \mathbf{z}_S^k)$ là hàm của trạng thái ước lượng robot tại thời điểm k và $k + 1$, đồng thời là hàm của tất cả các giá trị đo IMU từ k đến $k + 1$ tổng hợp thành $\mathbf{z}_S^k$. Giả sử $\mathbf{e}_{\text{imu}}^k$ tuân theo phân phối Gaussian:

$$f(\mathbf{e}_{\text{imu}}^k | \mathbf{x}_R^k, \mathbf{x}_R^{k+1}) \approx \mathcal{N}(\mathbf{0}, \mathcal{R}_{\text{imu}}^k) \quad (3.34)$$

Để dự báo $\hat{\mathbf{x}}_R^{k+1}(\mathbf{x}_R^k, \mathbf{z}_S^k)$ và covariance tương ứng $\mathbf{P}(\delta \hat{\mathbf{x}}_R^{k+1} | \mathbf{x}_R^k, \mathbf{z}_S^k)$, ta cần rời rạc hóa mô hình tuyến tính hóa IMU tại (3.18). Áp dụng phương pháp Runge-Kutta, ta

được phương trình trạng thái phi tuyến thời gian rời rạc $\mathbf{f}_d(\bar{\mathbf{x}}_R^k)$ và ma trận trạng thái tuyến tính hóa $\mathbf{F}_d(\bar{\mathbf{x}}_R^k)$. Từ đó sai số IMU được tính bằng:

$$\mathbf{e}_{\text{imu}}^k(\mathbf{x}_R^k, \mathbf{x}_R^{k+1}, \mathbf{z}_s^k) = \begin{bmatrix} \hat{\mathbf{p}}_{WI}^{k+1} - \mathbf{p}_{WI}^{k+1} \\ 2[\hat{\mathbf{q}}_{WI}^{k+1} \otimes (\mathbf{q}_{WI}^{k+1})^{-1}]_{1:3} \\ \hat{\mathbf{x}}_{\text{sb}}^{k+1} - \mathbf{x}_{\text{sb}}^{k+1} \end{bmatrix} \in \mathbb{R}^{15} \quad (3.35)$$

Thành lập ma trận trọng số IMU là nghịch đảo của covariance nhiễu:

$$\mathbf{W}_{\text{imu}}^k = \mathcal{R}_{\text{imu}}^k{}^{-1} = \left(\frac{\partial \mathbf{e}_{\text{imu}}^k}{\partial \delta \hat{\mathbf{x}}_R^{k+1}} \mathbf{P}(\delta \mathbf{x}_R^{k+1} | \mathbf{x}_R^k, \mathbf{z}_s^k) \frac{\partial \mathbf{e}_{\text{imu}}^k}{\partial \delta \hat{\mathbf{x}}_R^{k+1}}{}^T \right)^{-1} \quad (3.36)$$

Các ma trận Jacobian của $\mathbf{e}_{\text{imu}}^k$ theo $\delta \hat{\mathbf{x}}_R^{k+1}$ và $\mathbf{x}_R^k$ lần lượt là:

$$\frac{\partial \mathbf{e}_{\text{imu}}^k}{\partial \delta \hat{\mathbf{x}}_R^{k+1}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 9} \\ \mathbf{0}_{3 \times 3} & [\hat{\mathbf{q}}_{WI}^{k+1} \otimes (\mathbf{q}_{WI}^{k+1})^{-1}]_{1:3,1:3}^\oplus & \mathbf{0}_{3 \times 9} \\ \mathbf{0}_{9 \times 3} & \mathbf{0}_{9 \times 3} & \mathbf{I}_9 \end{bmatrix} \quad (3.37)$$

$$\frac{\partial \mathbf{e}_{\text{imu}}^k}{\partial \delta \mathbf{x}_R^k} = \frac{\partial \mathbf{e}_{\text{imu}}^k}{\partial \delta \hat{\mathbf{x}}_R^{k+1}} \mathbf{F}_d(\bar{\mathbf{x}}_R^R, \Delta \mathbf{t}^R) \mathbf{F}_d(\bar{\mathbf{x}}_R^{R-1}, \Delta \mathbf{t}^{R-1}) \dots \mathbf{F}_d(\bar{\mathbf{x}}_R^k, \Delta \mathbf{t}^0) \quad (3.38)$$

CHƯƠNG 4. HỆ THỐNG ĐỊNH VỊ TÍCH HỢP THỊ GIÁC LẬP THỂ, QUÁN TÍNH VÀ GPS

4.1. Căn chỉnh hệ thống VIO/GPS

4.1.1. Thuật toán căn chỉnh tọa độ

Trước khi thực hiện tích hợp, ta cần tìm phép biến đổi giữa ngõ ra VIO và ngõ ra GPS nhằm đưa chúng về cùng một quan hệ tọa độ. Các hệ tọa độ được sử dụng trong bài toán gồm: $\mathcal{F}_I$, hệ tọa độ gắn với IMU đồng thời là tâm hệ VIO; $\mathcal{F}_W$, hệ tọa độ thế giới khởi tạo bởi thuật toán VIO; $\mathcal{F}_A$, hệ tọa độ gắn với anten GPS; và $\mathcal{F}_G$, hệ tọa độ ENU gắn với Trái Đất. Ta có ngõ ra hệ VIO $\mathbf{T}_{WI}$ thể hiện tư thế của $\mathcal{F}_I$ trong $\mathcal{F}_W$, và ngõ ra hệ GPS $\mathbf{T}_{GA}$ thể hiện tư thế của $\mathcal{F}_A$ trong $\mathcal{F}_G$. Xét phép biến đổi:

$$\mathbf{T}_{GA} = \mathbf{T}_{GW} \mathbf{T}_{WI} \mathbf{T}_{IA} \quad (4.1)$$

Ta thấy nếu biết giá trị của $\mathbf{T}_{GW}$ và $\mathbf{T}_{IA}$, ngõ ra VIO $\mathbf{T}_{WI}^{\text{vio}}$ có thể chuyển sang dạng tọa độ GPS $\mathbf{T}_{GA}^{\text{gps}}$ và ngược lại. Đối với $\mathbf{T}_{IA}$, ta đo thực tế khoảng cách $\mathbf{t}_{IA}$ từ tâm IMU đến anten GPS bằng thước milimet, và giả sử $\mathbf{R}_{IA} = \mathbf{I}_3$ do anten GPS là vô hướng. Trong khi đó, $\mathbf{T}_{GW}$ cần được tính ngược lại từ ngõ ra hai hệ. Với mỗi cặp tư thế VIO và GPS tại thời điểm t , ta có:

$$\mathbf{T}_{GA}^{\text{gps}}(t) = \mathbf{T}_{GW} \mathbf{T}_{WI}^{\text{vio}}(t) \mathbf{T}_{IA} = \mathbf{T}_{GW} \mathbf{T}_{WA}^{\text{vio}}(t) \quad (4.2)$$

$$\begin{bmatrix} \mathbf{R}_{GA}^{\text{gps}}(t) & \mathbf{p}_{SA}^{\text{gps}}(t) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{GW} & \mathbf{t}_{GW} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{WA}^{\text{vio}}(t) & \mathbf{p}_{WA}^{\text{vio}}(t) \\ 0 & 1 \end{bmatrix} \quad (4.3)$$

$$\begin{cases} \mathbf{R}_{GA}^{\text{gps}}(t) = \mathbf{R}_{GW} \mathbf{R}_{WA}^{\text{vio}}(t) \end{cases} \quad (4.4)$$

$$\begin{cases} \mathbf{p}_{GA}^{\text{gps}}(t) = \mathbf{R}_{GW} \mathbf{p}_{WA}^{\text{vio}}(t) + \mathbf{t}_{GW} \end{cases} \quad (4.5)$$

Vì hệ GPS không quan sát được hướng $\mathbf{R}_{GA}^{\text{gps}}$ nên ta không thể sử dụng phương trình (4.4) để giải $\mathbf{R}_{GW}$. Thay vào đó, ta cần tìm đồng thời $\mathbf{R}_{GW}$ và $\mathbf{t}_{GW}$ từ phương trình (4.5). Vì phép biến đổi $\mathbf{T}_{GW}$ là cố định nên có thể lấy hai chuỗi dữ liệu $\mathbf{p}_{GA}^{\text{gps}}$ và $\mathbf{p}_{WA}^{\text{vio}}$ theo thời gian để tìm nghiệm tối ưu theo phương pháp bình phương cực tiểu. Ta phát biểu lại bài toán như sau:

Cho $\mathcal{A} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\}$ và $\mathcal{B} = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n\}$ là hai tập hợp điểm tương ứng nhau trong không gian. Tìm phép xoay và tịnh tiến $(\mathbf{R}, \mathbf{t})$ biến đổi tập $\mathcal{A}$ thành $\mathcal{B}$ sao cho hàm mục tiêu tổng bình phương khoảng cách giữa các điểm tương ứng đạt giá trị nhỏ nhất:

$$J(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^n \|(\mathbf{R}\mathbf{a}_i + \mathbf{t}) - \mathbf{b}_i\|^2 \quad (4.6)$$

Trước tiên, ta đạo hàm J để tìm cực tiểu theo $\mathbf{t}$:

$$0 = \frac{\partial J}{\partial \mathbf{t}} = 2 \sum_{i=1}^n (\mathbf{R}\mathbf{a}_i + \mathbf{t} - \mathbf{b}_i) \quad (4.7)$$

$$\Leftrightarrow \mathbf{R} \sum_{i=1}^n \mathbf{a}_i + n\mathbf{t} - \sum_{i=1}^n \mathbf{b}_i = 0 \quad (4.8)$$

$$\Leftrightarrow \mathbf{t} = \frac{1}{n} \sum_{i=1}^n \mathbf{b}_i - \mathbf{R} \frac{1}{n} \sum_{i=1}^n \mathbf{a}_i$$

Đặt $\bar{\mathbf{a}}, \bar{\mathbf{b}}$ là tâm trung bình của mỗi tập $\mathcal{A}$ và $\mathcal{B}$:

$$\bar{\mathbf{a}} \triangleq \frac{1}{n} \sum_{i=1}^n \mathbf{a}_i, \quad \bar{\mathbf{b}} \triangleq \frac{1}{n} \sum_{i=1}^n \mathbf{b}_i \quad (4.9)$$

Suy ra:

$$\mathbf{t} = \bar{\mathbf{b}} - \mathbf{R}\bar{\mathbf{a}} \quad (4.10)$$

Thay $\mathbf{t}$ vào (4.6) và đặt $\mathbf{x}_i \triangleq \mathbf{a}_i - \bar{\mathbf{a}}, \mathbf{y}_i \triangleq \mathbf{b}_i - \bar{\mathbf{b}}$, bài toán được đơn giản về việc tìm phép xoay $\mathbf{R}$ giữa hai tập điểm $\mathbf{x}_i$ và $\mathbf{y}_i$ có thành phần tịnh tiến bằng 0.

$$J(\mathbf{R}) = \sum_{i=1}^n \|\mathbf{R}(\mathbf{a}_i - \bar{\mathbf{a}}) - (\mathbf{b}_i - \bar{\mathbf{b}})\|^2 = \sum_{i=1}^n \|\mathbf{R}\mathbf{x}_i - \mathbf{y}_i\|^2 \quad (4.11)$$

$$\mathbf{R} = \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{i=1}^n \|\mathbf{R}\mathbf{x}_i - \mathbf{y}_i\|^2 \quad (4.12)$$

$$\begin{aligned}
 &= \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{i=1}^n (\mathbf{x}_i \mathbf{x}_i^T - 2\mathbf{y}_i^T \mathbf{R} \mathbf{x}_i + \mathbf{y}_i \mathbf{y}_i^T) \\
 &= \underset{\mathbf{R}}{\operatorname{argmin}} \left(\sum_{i=1}^n -2\mathbf{y}_i^T \mathbf{R} \mathbf{x}_i \right) = \underset{\mathbf{R}}{\operatorname{argmax}} \left(\sum_{i=1}^n \mathbf{y}_i^T \mathbf{R} \mathbf{x}_i \right)
 \end{aligned}$$

Đặt hai ma trận dữ liệu $\mathbf{X} \triangleq [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_n]$, $\mathbf{Y} \triangleq [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \dots \quad \mathbf{y}_n]$ chứa tất cả các vector điểm của hai tập hợp, đồng thời áp dụng toán tử vết ma trận $\operatorname{tr}(\cdot)$, ta có:

$$\sum_{i=1}^n \mathbf{y}_i^T \mathbf{R} \mathbf{x}_i = \operatorname{tr}(\mathbf{Y}^T \mathbf{R} \mathbf{X}) = \operatorname{tr}(\mathbf{R} \mathbf{X} \mathbf{Y}^T) \quad (4.13)$$

Tiếp tục đặt $\mathbf{S} \triangleq \mathbf{X} \mathbf{Y}^T$, hàm cần cực đại trở thành:

$$\operatorname{tr}(\mathbf{R} \mathbf{X} \mathbf{Y}^T) = \operatorname{tr}(\mathbf{R} \mathbf{S}) \quad (4.14)$$

Đối với trường hợp phép xoay 3 trục tổng quát, ta có thể tìm $\mathbf{R}$ tối ưu bằng cách phân tích ma trận $\mathbf{S}$ theo phương pháp SVD (Singular Value Decomposition) [22]. Tuy nhiên, hệ thống VIO trong đề tài có lợi thế là quan sát được hai góc roll, pitch tuyệt đối, từ đó đã căn chỉnh trục z của $\mathcal{F}_W$ cùng phương với trục z của $\mathcal{F}_G$ và vector trọng trường. Nhờ vậy, phép xoay cần tìm chỉ còn là phép xoay 1 góc yaw. Việc không căn chỉnh roll và pitch trong bước này giúp ta tránh được sai số gây ra bởi nhiễu lớn trên phép đo cao độ GPS. Bên cạnh đó, việc giảm số ẩn cần giải giúp ta hạn chế được một số trường hợp suy biến so với lời giải tổng quát, ví dụ như khi quỹ đạo của vật là một đường thẳng.

Bài toán thu gọn về việc tìm góc xoay yaw ψ sao cho:

$$\begin{aligned}
 \psi &= \underset{\psi}{\operatorname{argmax}} \left(\operatorname{tr}(\mathbf{R}_{z,\psi} \mathbf{S}) \right) \\
 &= \underset{\psi}{\operatorname{argmax}} \left(\operatorname{tr} \left(\begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \right) \right) \\
 &= \underset{\psi}{\operatorname{argmax}} \left((S_{12} - S_{21})s_\psi + (S_{11} + S_{22})c_\psi + S_{33} \right)
 \end{aligned} \quad (4.15)$$

$$= \operatorname{argmax}_{\psi} \left((S_{12} - S_{21})s_{\psi} + (S_{11} + S_{22})c_{\psi} \right)$$

Đặt $u \triangleq S_{12} - S_{21}$, $v \triangleq S_{11} + S_{22}$ và định nghĩa góc α sao cho $c_{\alpha} \triangleq \frac{u}{\sqrt{u^2+v^2}}$, $s_{\alpha} \triangleq \frac{v}{\sqrt{u^2+v^2}}$, góc ψ cần tìm là:

$$\psi = \operatorname{argmax}_{\psi} (\sin(\alpha + \psi)) = \frac{\pi}{2} - \operatorname{atan2}(S_{11} + S_{22}, S_{12} - S_{21}) \quad (4.16)$$

Suy ra nghiệm bài toán:

$$\mathbf{R} = \mathbf{R}_{z,\psi}, \quad \mathbf{t} = \bar{\mathbf{b}} - \mathbf{R}\bar{\mathbf{a}} \quad (4.17)$$

Thuật toán căn chỉnh được tóm tắt trong bảng sau.

Bảng 4.1. Tóm tắt thuật toán căn chỉnh với 1 góc xoay yaw

Tìm phép biến đổi $(\mathbf{R}, \mathbf{t})$ sao cho cực tiểu hàm mục tiêu:

$$J = \sum_{i=1}^n \|(\mathbf{R}\mathbf{a}_i + \mathbf{t}) - \mathbf{b}_i\|^2$$

1. Tính tâm trung bình của mỗi tập hợp điểm:

$$\bar{\mathbf{a}} = \frac{1}{n} \sum_{i=1}^n \mathbf{a}_i, \quad \bar{\mathbf{b}} = \frac{1}{n} \sum_{i=1}^n \mathbf{b}_i$$

2. Tính các vector tịnh tiến so với tâm:

$$\mathbf{x}_i = \mathbf{a}_i - \bar{\mathbf{a}}, \quad \mathbf{y}_i = \mathbf{b}_i - \bar{\mathbf{b}}$$

3. Tính ma trận dữ liệu vào:

$$\mathbf{S} = \mathbf{XY}^T$$

4. Tính góc xoay yaw tối ưu:

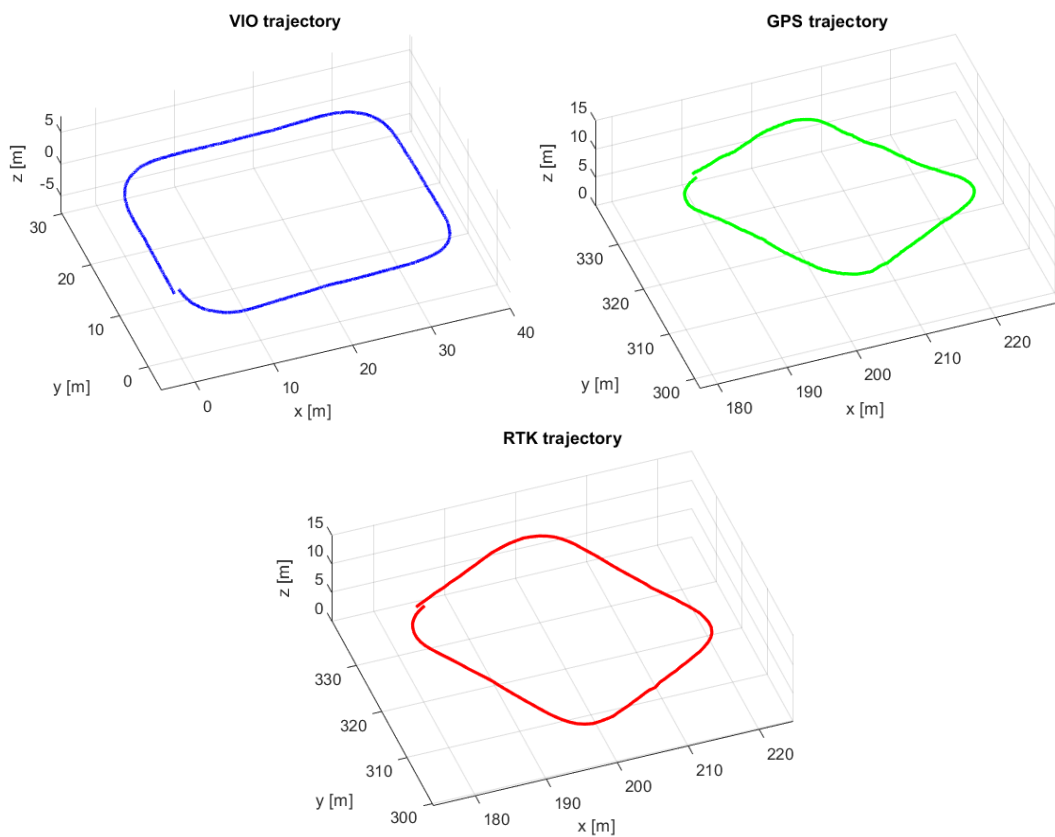
$$\psi = \frac{\pi}{2} - \operatorname{atan2}(S_{11} + S_{22}, S_{12} - S_{21})$$

5. Tính phép xoay và tịnh tiến tối ưu:

$$\mathbf{R} = \mathbf{R}_{z,\psi}, \quad \mathbf{t} = \bar{\mathbf{b}} - \mathbf{R}\bar{\mathbf{a}}$$

4.1.2. Kết quả thực nghiệm căn chỉnh hệ VIO/GPS

Nhằm đánh giá giải thuật đề xuất, học viên thử nghiệm căn chỉnh quỹ đạo VIO và GPS theo hai phương pháp Yaw và SVD, sau đó so sánh kết quả với hệ đo động thời gian thực (RTK) cấp chính xác centimét. Cả ba hệ được gắn cố định với nhau, di chuyển trên mặt đất theo quỹ đạo hình chữ nhật kích thước 35x30m như Hình 4.1. Thời gian đi hết quãng đường là 112s, trong đó 30s đầu tiên được dùng làm dữ liệu căn chỉnh. Quỹ đạo RTK được biểu diễn trong hệ tọa độ ENU với gốc tọa độ trùng với GPS.

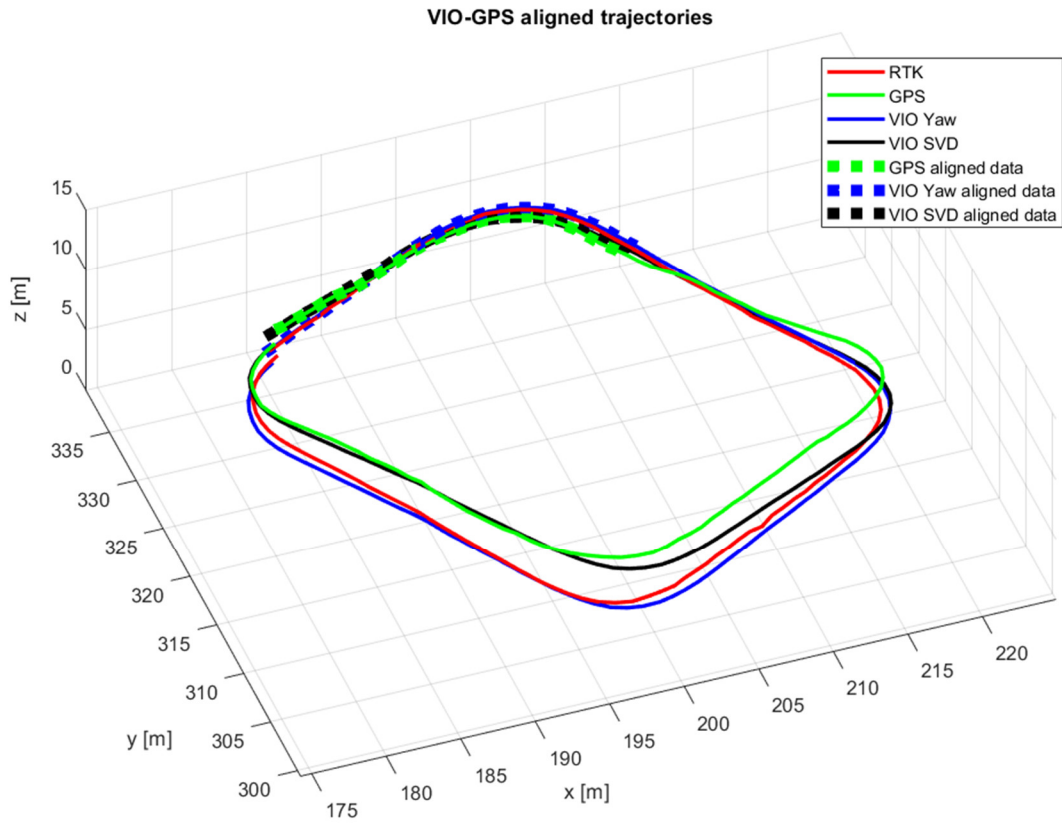


Hình 4.1 Quỹ đạo VIO, GPS ban đầu và quỹ đạo chuẩn RTK

Hình 4.2 cho thấy quỹ đạo VIO sau khi căn chỉnh sang hệ tọa độ GPS. Đường màu đỏ thể hiện quỹ đạo chuẩn RTK, đường màu xanh lá thể hiện quỹ đạo GPS, đường màu xanh dương và màu đen lần lượt là quỹ đạo VIO sau khi căn chỉnh theo phương pháp Yaw và SVD. Các đường đứt nét kí hiệu dữ liệu căn chỉnh. Giữa hai phương pháp căn chỉnh, ta thấy quỹ đạo VIO Yaw bám sát và đồng phẳng hơn theo quỹ đạo chuẩn RTK, trong khi quỹ đạo VIO SVD có xu hướng nghiêng theo sai lệch cao độ của GPS. Lưu ý là cả hai phương pháp đều sử dụng dữ liệu căn chỉnh là GPS,

và chỉ dùng RTK để đánh giá sai số. Phương pháp SVD đã thực hiện đúng chức năng là tìm phép xoay 3 trục để quỹ đạo VIO bám tối ưu theo GPS. Tuy nhiên trong điều kiện thực tế, nhiễu lớn trên cao độ GPS sẽ làm sai lệch kết quả tính toán 2 góc roll pitch, khiến cho quỹ đạo căn chỉnh bị nghiêng và không khớp với quỹ đạo chuẩn. Trong trường hợp này, với giả định hai góc roll pitch đã được xác định đúng bởi hệ VIO, phương pháp Yaw chỉ cần tìm phép xoay quanh trục z từ dữ liệu vị trí x-y nhiễu nhỏ, nhờ vậy đạt được độ chính xác cao hơn, như Hình 4.3 và Bảng 4.2.

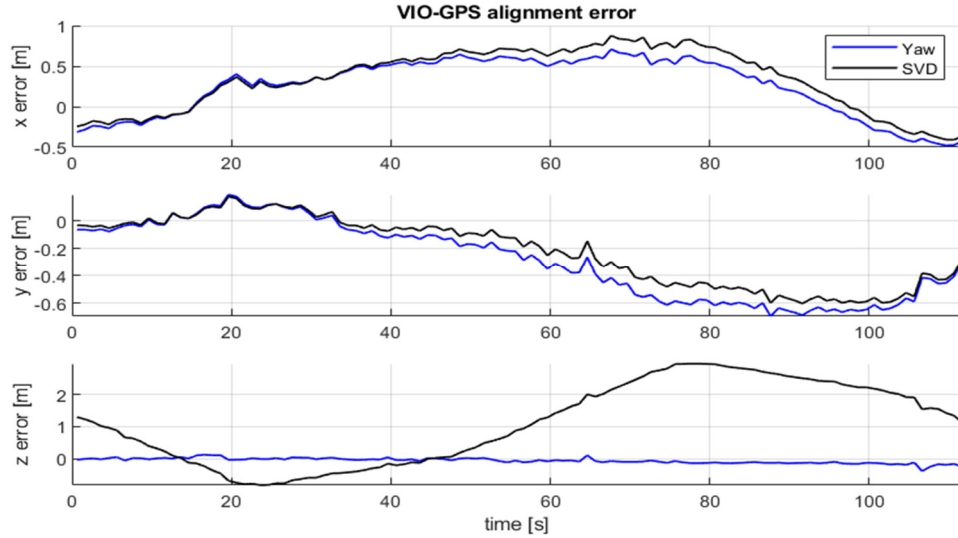
Như vậy, phương pháp căn chỉnh chỉ xoay góc yaw phù hợp với hệ thống VIO/GPS trong đề tài hơn phương pháp xoay cả ba trục theo SVD.



Hình 4.2 Các quỹ đạo sau khi căn chỉnh

Phương pháp	Độ dời [m]			Góc xoay [°]			Sai số RMS vị trí [m]	
	x_{GW}	y_{GW}	z_{GW}	φ_{GW}	θ_{GW}	ψ_{GW}	e_{rms2D}	e_{rms3D}
SVD	184.1	328.5	7.64	-5.86	-2.42	-61.25	0.604	1.774
Yaw	184.1	328.5	9.36	0	0	-61.53	0.586	0.595

Bảng 4.2 Kết quả tọa độ và sai số căn chỉnh



Hình 4.3 Sai số căn chỉnh so với quỹ đạo chuẩn theo thời gian

4.2. Thuật toán định vị tích hợp VIO/GPS

4.2.1. Bộ lọc Kalman mở rộng EKF

Bộ lọc Kalman là một thuật toán ước lượng trạng thái tối ưu dựa trên mô hình toán, ngõ vào và ngõ ra của hệ thống theo thời gian. Điểm nổi bật của bộ lọc Kalman là khả năng kết hợp nhiều tín hiệu vào ra với độ không chắc chắn khác nhau để cho ra kết quả ước lượng tốt nhất. So với các bộ lọc theo tần số, bộ lọc Kalman có khả năng khử nhiễu hiệu quả mà không làm trễ tín hiệu. Bên cạnh đó, nhờ tính chất đệ quy, trạng thái hiện tại không phụ thuộc vào quá khứ xa hơn 1 chu kỳ, bộ lọc Kalman có tốc độ tính toán nhanh, tiết kiệm tài nguyên xử lý và được sử dụng rộng rãi trong thực tế. Tuy nhiên, bộ lọc Kalman cơ bản chỉ giới hạn cho các đối tượng tuyến tính. Đối với các đối tượng phi tuyến như hệ thống định vị, giải pháp là tuyến tính hóa hệ thống quanh giá trị ước lượng tại mỗi chu kỳ, rồi áp dụng quy trình Kalman lên mô hình xấp xỉ. Đó cũng là nguyên tắc của bộ lọc Kalman mở rộng (EKF) [23] được trình bày dưới đây.

Cho một hệ thống phi tuyến mô tả bởi hệ phương trình thời gian rời rạc:

$$\begin{cases} \mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{v}_k) \end{cases} \quad (4.18)$$

trong đó $\mathbf{x}_k$ là vector trạng thái, $\mathbf{u}_k$ là tín hiệu vào, $\mathbf{z}_k$ là tín hiệu ra đo được tại thời điểm k , $\mathbf{f}(\cdot)$ và $\mathbf{h}(\cdot)$ là các hàm phi tuyến biểu diễn mô hình quá trình và mô hình đo lường của hệ thống, $\mathbf{w}_k$ và $\mathbf{v}_k$ là các nhiễu trắng Gaussian tác động lên $\mathbf{f}(\cdot)$ và $\mathbf{h}(\cdot)$ với covariance tương ứng là $\mathbf{Q}_k$ và $\mathbf{R}_k$.

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k), \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k) \quad (4.19)$$

Quy trình thực hiện bộ lọc EKF được mô tả trong Bảng 4.3.

Bảng 4.3. Bộ lọc EKF rời rạc

Khởi tạo trạng thái $\hat{\mathbf{x}}_0$ và covariance sai số $\mathbf{P}_0$. $\hat{\mathbf{x}}_0 = \mathbf{E}(\mathbf{x}_0), \quad \mathbf{P}_0 = \mathbf{E}[(\mathbf{x} - \hat{\mathbf{x}}_0)(\mathbf{x} - \hat{\mathbf{x}}_0)^T]$ Tại mỗi chu kì lấy mẫu $k = 1, 2, \dots$, lặp lại các bước sau:
1. Dự đoán: (a) Dự đoán trạng thái tiên nghiệm: $\hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k)$ (b) Tính ma trận Jacobian của mô hình quá trình: $\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right _{\hat{\mathbf{x}}_{k-1}}$ (c) Dự đoán covariance sai số tiên nghiệm: $\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1}$
2. Cập nhật: (a) Dự đoán giá trị đo: $\hat{\mathbf{z}}_k = \mathbf{h}(\hat{\mathbf{x}}_k^-)$ (b) Tính ma trận Jacobian của mô hình đo lường: $\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k}{\partial \mathbf{x}} \right _{\hat{\mathbf{x}}_k^-}$ (c) Tính độ lợi Kalman tối ưu: $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$

(d) Cập nhật trạng thái và covariance sai số hậu nghiệm:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

4.2.2. Mô hình toán hệ thống định vị tích hợp VIO/GPS

Phần này tập trung vào xây dựng mô hình toán của hệ tích hợp lỏng VIO/GPS theo hướng áp dụng bộ lọc EKF, trong đó kết quả VIO được dùng để dự đoán và kết quả GPS được dùng để cập nhật các trạng thái ước lượng. Mặc dù việc xây dựng mô hình cập nhật GPS trên cả 6 trục là hoàn toàn khả thi về mặt lý thuyết, trong đề tài này học viên chọn mô hình cập nhật 4 trục x, y, z, yaw vì hai lý do. Thứ nhất, góc roll và pitch ước lượng từ hệ VIO không chứa sai số tích lũy nhờ việc quan sát gia tốc trọng trường trên IMU. Thứ hai, việc cập nhật roll pitch phải sử dụng giá trị đo trục z của GPS thường bị nhiễu rất lớn, khiến cho kết quả roll pitch sau khi tích hợp nếu có cũng kém hơn ban đầu.

Định nghĩa vector trạng thái gồm vị trí $\mathbf{p}_{WI}$ và góc yaw ψ_{WI} theo hệ VIO:

$$\mathbf{x} \triangleq [\mathbf{p}_{WI}^T \quad \psi_{WI}]^T \quad (4.20)$$

Mô hình quá trình của hệ tích hợp tại thời điểm cập nhật VIO k :

$$\begin{cases} \mathbf{p}_{WI k} = \mathbf{p}_{WI k-1} + \mathbf{R}_{WI k-1} \delta \mathbf{p}_{WI k}^{\text{vio}} + \mathbf{w}_{p k-1} \\ \psi_{WI k} = \psi(\mathbf{R}_{WI k-1} \delta \mathbf{R}_{WI k}^{\text{vio}}) + w_{\psi k-1} = \psi(\mathbf{R}_{WI k}) + w_{\psi k-1} \end{cases} \quad (4.21)$$

trong đó các ngõ vào $\delta \mathbf{p}_{WI k}^{\text{vio}}$, $\delta \mathbf{R}_{WI k}^{\text{vio}}$ là chênh lệch vị trí và hướng của hệ VIO giữa thời điểm k và $k - 1$:

$$\begin{cases} \delta \mathbf{p}_{WI k}^{\text{vio}} \triangleq (\mathbf{R}_{WI k-1}^{\text{vio}})^T (\mathbf{p}_{WI k}^{\text{vio}} - \mathbf{p}_{WI k-1}^{\text{vio}}) \\ \delta \mathbf{R}_{WI k}^{\text{vio}} \triangleq (\mathbf{R}_{WI k-1}^{\text{vio}})^T \mathbf{R}_{WI k}^{\text{vio}} \end{cases} \quad (4.22)$$

$\mathbf{w}_{p k-1}$, $w_{\psi k-1}$ lần lượt là nhiễu trắng Gaussian trên số gia vị trí và góc hướng yaw với covariance $\mathbf{Q}_{k-1}$. Hàm $\psi(\mathbf{R})$, bên cạnh $\theta(\mathbf{R})$ và $\varphi(\mathbf{R})$, là các hàm xác định góc xoay yaw, pitch, roll từ ma trận xoay $\mathbf{R}$:

$$\psi(\mathbf{R}) = \text{atan2}(r_{21}, r_{11}) \quad (4.23)$$

$$\theta(\mathbf{R}) = \text{atan2}\left(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2}\right) \quad (4.24)$$

$$\varphi(\mathbf{R}) = \text{atan2}(r_{32}, r_{33}) \quad (4.25)$$

Nhận xét về mô hình quá trình trên, ta chọn ngõ vào là các số gia $\delta \mathbf{p}_{WI}^{\text{vio}}, \delta \mathbf{R}_{WI}^{\text{vio}}$ thay vì bản thân vị trí và hướng của VIO nhằm mục đích khử sai số trên số gia ngay khi xuất hiện, ngăn sai số tích lũy đi vào hệ tích hợp. Mặt khác, tuy chỉ cập nhật góc xoay yaw trong $\mathbb{R}$, ta vẫn lan truyền phép xoay trong $SO(3)$ theo công thức $\mathbf{R}_{WI_k} = \mathbf{R}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}}$, thay vì $\psi_{WI_k} = \psi_{WI_{k-1}} + \delta \psi_{WI_k}^{\text{vio}}$. Điều này giúp giữ nguyên tính chính xác của phép xoay trong tập xác định gốc.

Mô hình đo lường của hệ tại thời điểm cập nhật GPS n :

$$\mathbf{p}_{GA_n}^{\text{gps}} = \mathbf{R}_{GW}(\mathbf{R}_{WI_n} \mathbf{t}_{IA} + \mathbf{p}_{WI_n}) + \mathbf{t}_{GW} + \mathbf{v}_n \quad (4.26)$$

trong đó $\mathbf{v}_n$ là nhiễu Gaussian của phép đo GPS với ma trận covariance nhiễu $\mathbf{R}_n$. $\mathbf{R}_{GW}, \mathbf{t}_{GW}$ là phép xoay và tịnh tiến thu được từ quá trình cân chỉnh VIO/GPS. $\mathbf{t}_{IA}$ là khoảng cách offset từ IMU đến anten GPS được đo đạc từ hệ thống thực tế.

Do VIO và GPS có tốc độ và thời điểm cập nhật khác nhau, ta sẽ sử dụng phương pháp cập nhật bất đồng bộ cho bộ ước lượng. Xét thời điểm cập nhật GPS n nằm giữa hai mẫu VIO gần nhất là k_1 và k_2 , ta sẽ nội suy trạng thái hệ thống từ k_1 đến n để thực hiện việc cập nhật GPS, sau đó nội suy từ n đến k_2 để tiếp tục vòng cập nhật VIO. Vector vị trí được nội suy tuyến tính. Trong khi đó, phép xoay được nội suy thông qua quaternion sử dụng phương pháp Slerp (Spherical linear interpolation) [24]:

$$\begin{aligned} \tau &= (t_n - t_{k_1}) / (t_{k_2} - t_{k_1}) \in [0, 1] \\ \mathbf{p}_n &= (\mathbf{p}_{k_2} - \mathbf{p}_{k_1})\tau + \mathbf{p}_{k_1} \\ \mathbf{q}_n &= \mathbf{q}_{k_1} \otimes (\mathbf{q}_{k_1}^{-1} \otimes \mathbf{q}_{k_2})^\tau \end{aligned} \quad (4.27)$$

4.2.3. Thuật toán định vị tích hợp VIO/GPS sử dụng bộ lọc EKF

Ta triển khai bộ lọc EKF lên mô hình tích hợp VIO/GPS theo các bước sau đây.

Khởi tạo: Cho $\hat{\mathbf{x}}_0 = [\mathbf{p}_{WI_0}^{\text{vio}T} \quad \psi_{WI_0}^{\text{vio}}]^T$, $\hat{\mathbf{R}}_{WI_0} = \mathbf{R}_{WI_0}^{\text{vio}}$, $\mathbf{P}_0 = \mathbf{0}_{4 \times 4}$.

Dự đoán: Tại mỗi thời điểm cập nhật VIO k , ta dự đoán trạng thái tiên nghiệm theo phương trình (4.21):

$$\begin{cases} \hat{\mathbf{p}}_{WI_k}^- = \hat{\mathbf{p}}_{WI_{k-1}} + \hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{p}_{WI_k}^{\text{vio}} \\ \hat{\psi}_{WI_k}^- = \psi(\hat{\mathbf{R}}_{WI_k}^-) = \psi(\hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}}) \end{cases} \quad (4.28)$$

Tuyến tính hóa mô hình quá trình:

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}} = \begin{bmatrix} \frac{\partial \hat{\mathbf{p}}_{WI_k}}{\partial \hat{\mathbf{p}}_{WI_{k-1}}} & \frac{\partial \hat{\mathbf{p}}_{WI_k}}{\partial \hat{\psi}_{WI_{k-1}}} \\ \frac{\partial \hat{\psi}_{WI_k}}{\partial \hat{\mathbf{p}}_{WI_{k-1}}} & \frac{\partial \hat{\psi}_{WI_k}}{\partial \hat{\psi}_{WI_{k-1}}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{F}_{12} \\ \mathbf{0}_{1 \times 3} & \mathbf{F}_{22} \end{bmatrix} \quad (4.29)$$

Gọi $\mathbf{e}_z = [0 \quad 0 \quad 1]^T$ là vector đơn vị trục z , ta có tính chất:

$$\frac{\partial \mathbf{R}}{\partial \psi} = \frac{\partial \mathbf{R}_\psi}{\partial \psi} \mathbf{R}_\theta \mathbf{R}_\phi = [\mathbf{e}_z]^\times \mathbf{R}_\psi \mathbf{R}_\theta \mathbf{R}_\phi = [\mathbf{e}_z]^\times \mathbf{R} \quad (4.30)$$

Do đó:

$$\mathbf{F}_{12} = \frac{\partial}{\partial \psi_{WI_{k-1}}} (\hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{p}_{WI_k}^{\text{vio}}) = [\mathbf{e}_z]^\times \hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{p}_{WI_k}^{\text{vio}} \quad (4.31)$$

$$\mathbf{F}_{22} = \frac{\partial \psi(\hat{\mathbf{R}}_{WI_k})}{\partial \psi_{WI_{k-1}}} = \frac{\partial \text{atan2}(r_{21}(\hat{\mathbf{R}}_{WI_k}^-), r_{11}(\hat{\mathbf{R}}_{WI_k}^-))}{\partial \psi_{WI_{k-1}}} \quad (4.32)$$

trong đó $r_{11}(\mathbf{R}), r_{21}(\mathbf{R})$ là các hàm lấy phần tử của $\mathbf{R}$ tại vị trí (1,1) và (2,1).

Đạo hàm riêng của hàm $\text{atan2}(y, x)$ theo một lượng vô hướng a được cho bởi:

$$\frac{\partial \text{atan2}(y, x)}{\partial a} = \frac{x}{x^2 + y^2} \cdot \frac{\partial y}{\partial a} - \frac{y}{x^2 + y^2} \cdot \frac{\partial x}{\partial a} \quad (4.33)$$

Áp dụng vào (4.32) và đặt các hệ số $A = \frac{\hat{r}_{11}}{\hat{r}_{11}^2 + \hat{r}_{21}^2}$, $B = \frac{\hat{r}_{21}}{\hat{r}_{11}^2 + \hat{r}_{21}^2}$, ta tính được $\mathbf{F}_{22}$:

$$\begin{aligned}
 \mathbf{F}_{22} &= A \frac{\partial r_{21}(\hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}})}{\partial \psi_{WI_{k-1}}} - B \frac{\partial r_{11}(\hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}})}{\partial \psi_{WI_{k-1}}} \\
 &= Ar_{21} \left(\frac{\partial \hat{\mathbf{R}}_{WI_{k-1}}}{\partial \psi_{WI_{k-1}}} \delta \mathbf{R}_{WI_k}^{\text{vio}} \right) - Br_{11} \left(\frac{\partial \hat{\mathbf{R}}_{WI_{k-1}}}{\partial \psi_{WI_{k-1}}} \delta \mathbf{R}_{WI_k}^{\text{vio}} \right) \\
 &= Ar_{21}([\mathbf{e}_z]^\times \hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}}) - Br_{11}([\mathbf{e}_z]^\times \hat{\mathbf{R}}_{WI_{k-1}} \delta \mathbf{R}_{WI_k}^{\text{vio}})
 \end{aligned} \tag{4.34}$$

Từ $\mathbf{F}_{k-1}$ ta tính được covariance sai số tiên nghiệm $\mathbf{P}_k^-$:

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1} \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \tag{4.35}$$

Cập nhật: Do GPS và VIO có tần số cập nhật khác nhau, ta sẽ xét hai trường hợp (a)-không và (b)-có phép đo GPS n trong thời gian từ $k-1$ đến k .

Trong trường hợp (a), ta xem như độ không chắc chắn của phép đo GPS là vô cùng lớn $\mathbf{R}_k = \infty$, độ lợi Kalman $\mathbf{K}_k = 0$, dẫn đến trạng thái ước lượng và covariance hậu nghiệm đúng bằng tiên nghiệm $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-$, $\mathbf{P}_k = \mathbf{P}_k^-$.

Để cập nhật GPS trong trường hợp (b), ta nội suy trạng thái và covariance tiên nghiệm từ hai đầu mút $k-1$ và k đến n sử dụng công thức (4.27). Với $\hat{\mathbf{x}}_n^-$, $\mathbf{P}_n^-$ vừa nội suy, ta ước lượng giá trị đo GPS tại n theo (4.26):

$$\hat{\mathbf{p}}_{GA_n}^{\text{gps}} = \mathbf{R}_{GW}(\hat{\mathbf{R}}_{WI_n}^- \mathbf{t}_{IA} + \hat{\mathbf{p}}_{WI_n}^-) + \mathbf{t}_{GW} \tag{4.36}$$

Tuyến tính hóa mô hình đo:

$$\begin{aligned}
 \mathbf{H}_n &= \left. \frac{\partial \mathbf{h}_n}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_n^-} \\
 &= \left[\frac{\partial \hat{\mathbf{p}}_{GA_n}^{\text{gps}}}{\partial \hat{\mathbf{p}}_{WI_n}^-} \quad \frac{\partial \hat{\mathbf{p}}_{GA_n}^{\text{gps}}}{\partial \hat{\psi}_{WI_n}^-} \right] = \left[\mathbf{R}_{GW} \quad \frac{\partial (\mathbf{R}_{GW} \hat{\mathbf{R}}_{WI_n}^- \mathbf{t}_{IA})}{\partial \hat{\psi}_{WI_n}^-} \right] \\
 &= \left[\mathbf{R}_{GW} \quad \mathbf{R}_{GW} \frac{\partial (\hat{\mathbf{R}}_{WI_n}^-)}{\partial \hat{\psi}_{WI_n}^-} \mathbf{t}_{IA} \right] = \left[\mathbf{R}_{GW} \quad \mathbf{R}_{GW} [\mathbf{e}_z]^\times \hat{\mathbf{R}}_{WI_n}^- \mathbf{t}_{IA} \right]
 \end{aligned} \tag{4.37}$$

Ta thấy trong ma trận $\mathbf{H}_n$, không chỉ các thành phần vị trí (cột 1, 2, 3) mà thành phần góc yaw (cột 4) cũng có giá trị khác 0. Điều này nói lên thuật toán đề xuất có

khả năng cập nhật góc yaw chỉ bằng phép đo vị trí GPS, miễn là tồn tại khoảng cách offset $\mathbf{t}_{IA}$ giữa IMU và anten GPS.

Độ lợi Kalman được tính bằng:

$$\mathbf{K}_n = \mathbf{P}_n^- \mathbf{H}_n^T (\mathbf{H}_n \mathbf{P}_n^- \mathbf{H}_n^T + \mathbf{R}_n)^{-1} \quad (4.38)$$

Từ đó, trạng thái và covariance sai số hậu nghiệm được cập nhật:

$$\begin{bmatrix} \hat{\mathbf{p}}_{WI_n} \\ \hat{\psi}_{WI_n} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_{WI_n}^- \\ \hat{\psi}_{WI_n}^- \end{bmatrix} + \mathbf{K}_n (\mathbf{p}_{GA_n}^{\text{gps}} - \hat{\mathbf{p}}_{GA_n}^{\text{gps}}) \quad (4.39)$$

$$\mathbf{P}_n = (\mathbf{I} - \mathbf{K}_n \mathbf{H}_n) \mathbf{P}_n^- \quad (4.40)$$

Hướng xoay $SO(3)$ của hệ tích hợp tại thời điểm n được tính bằng tích giữa ma trận xoay yaw vừa cập nhật với các ma trận xoay pitch, roll dự đoán nội suy đến n :

$$\hat{\mathbf{R}}_{WI_n} = \mathbf{R}_\psi(\hat{\psi}_{WI_n}) \mathbf{R}_\theta(\hat{\theta}_{WI_n}^-) \mathbf{R}_\phi(\hat{\phi}_{WI_n}^-) \quad (4.41)$$

Sau khi cập nhật GPS hoàn tất, ta nội suy hệ thống quay lại thời điểm VIO k , tăng tiến $k \leftarrow k + 1$ và lặp lại quy trình dự đoán - cập nhật như trên.

Các covariance nhiễu $\mathbf{Q}$ và $\mathbf{R}$ được lấy online từ ngõ ra VIO và GPS để giúp hệ định vị tích hợp thích nghi theo chất lượng của các hệ con tại từng thời điểm.

CHƯƠNG 5. THỰC HIỆN VÀ ĐÁNH GIÁ HỆ THỐNG

5.1. Thực hiện hệ thống

5.1.1. Phần cứng

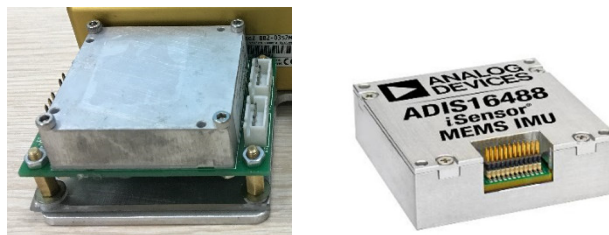
Các thiết bị sử dụng trong đề tài bao gồm:

Stereo camera Point Grey Bumblebee2 [25]: chụp đồng bộ cặp ảnh xám 640x480 pixel với tần số cài đặt là 20Hz và phát ra theo chuẩn FireWire (IEEE 1394a). Mỗi camera có tiêu cự 2.5mm, góc nhìn 97°. Baseline giữa hai camera là 12cm.



Hình 5.1. Stereo camera Point Grey Bumblebee2

IMU ADIS16488 [26]: đo gia tốc tịnh tiến, vận tốc xoay, từ trường trên 3 trục. IMU được đọc bởi một mạch xử lý dùng vi điều khiển STM32F4, thực hiện chuyển đổi dữ liệu sang chuẩn giao tiếp UART và phát ra với tần số tối đa là 500Hz. Mạch xử lý cũng chứa một thuật toán ước lượng nội góc xoay 3 trục, dùng làm dữ liệu chuẩn đánh giá hệ thống khi không có thiết bị ngoài. Bộ ước lượng có độ chính xác 0.1° với góc roll, pitch và 0.3° với góc yaw.



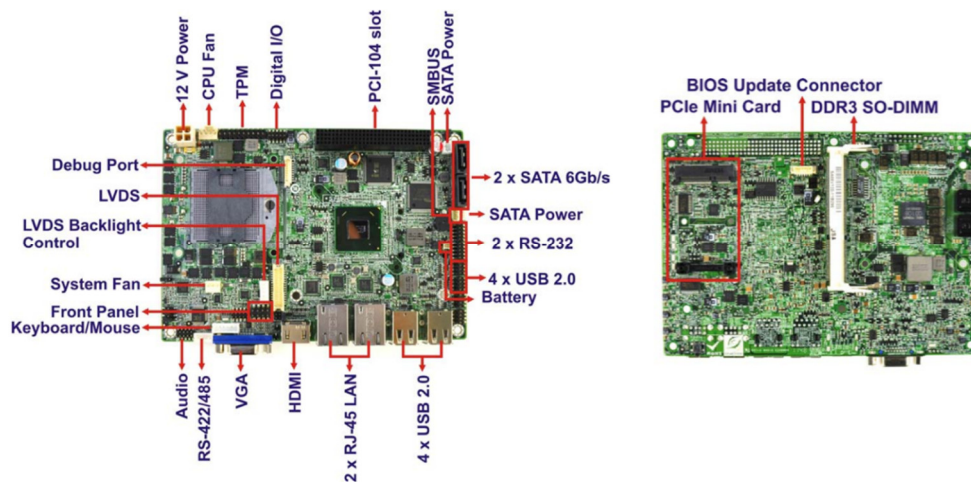
Hình 5.2. IMU ADIS16488 và mạch xử lý

Bộ thu GPS Ublox NEO-M8N [27]: đo vị trí kinh độ, vĩ độ, cao độ trên Trái Đất sử dụng cùng lúc hai hệ thống GPS và GLONASS với tần số cập nhật 5Hz. Độ chính xác phương ngang là 2.5m.



Hình 5.3. GPS Receiver Ublox NEO-M8N

Máy tính nhúng IEI NANO-HM650 [28]: bộ xử lý trung tâm của hệ thống, đọc dữ liệu từ các thiết bị và chạy giải thuật định vị tích hợp. Máy tính có kích thước nhỏ gọn 165x115mm, chạy CPU Core i7-2760MQ, có khe cắm card PCIe để chuyển đổi sang giao tiếp FireWire của stereo camera.



Hình 5.4. Máy tính nhúng IEI NANO-HM650

GPS RTK South S82T [29]: bộ định vị GPS vi sai với độ chính xác cấp cm, dùng làm vị trí chuẩn đánh giá hệ thống khi thử nghiệm ngoài trời.



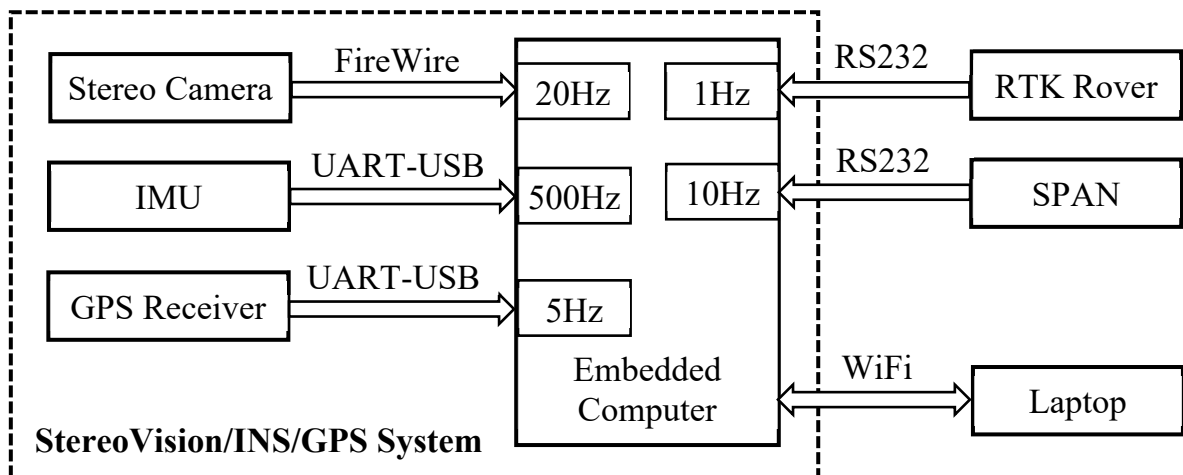
Hình 5.5. GPS RTK South S82T rover và base

Novatel SPAN-CPT [30]: bộ định vị 6 trục tích hợp GPS/INS/Wheel, sử dụng GPS hai tần số và gyroscope quang học. Góc xoay từ SPAN được dùng đánh giá hệ thống trong đề tài với độ chính xác 0.01° cho góc roll, pitch và 0.1° cho góc yaw. Vị trí từ SPAN cũng được dùng khi hệ RTK bị suy giảm do môi trường nhiều vật che chắn. Độ chính xác vị trí đơn điểm của SPAN là 1.2m.

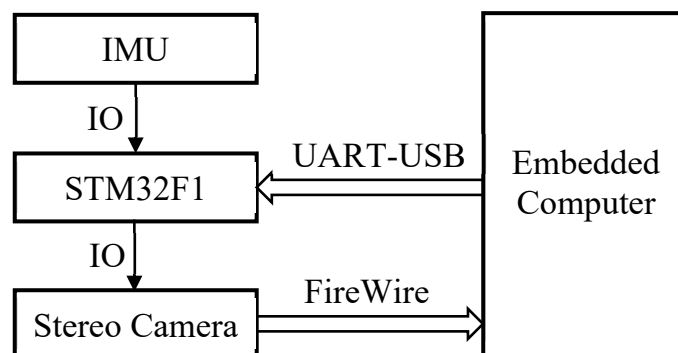


Hình 5.6. Novatel SPAN-CPT

Các thiết bị được kết nối theo sơ đồ bên dưới. Trong đó, một laptop được dùng để điều khiển và hiển thị kết quả định vị từ máy tính nhúng thông qua WiFi.

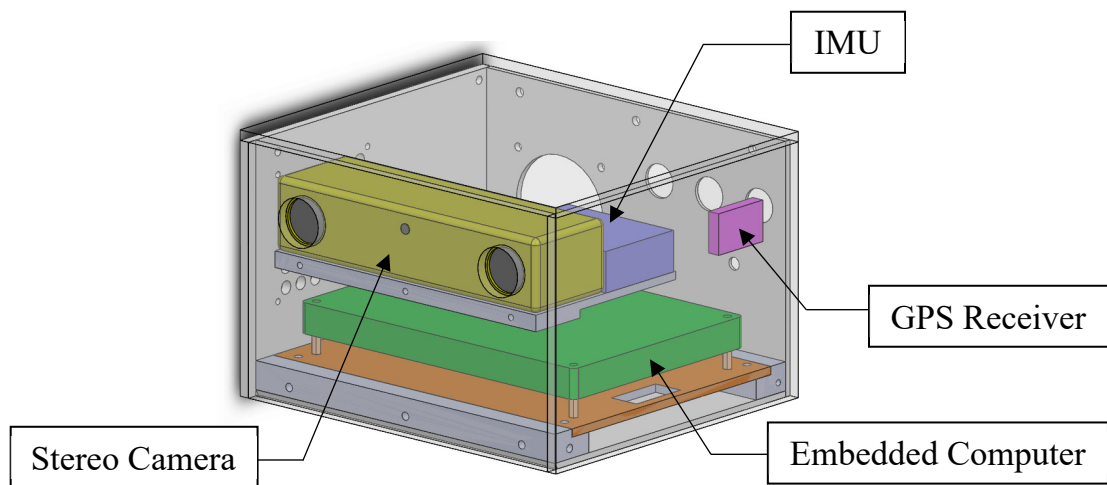


Hình 5.7. Sơ đồ phân cứng hệ thống StereoVision/INS/GPS và các thiết bị đánh giá



Hình 5.8. Sơ đồ phân cứng đồng bộ IMU-Camera

Sơ đồ Hình 5.8 mô tả việc đồng bộ phần cứng giữa IMU và camera theo phương pháp trong Chương 3. Theo đó, IMU phát ra xung clock đồng bộ với dữ liệu của nó. Vi điều khiển STM32F1 nhận xung clock, chia tần số và dịch pha để tạo ra xung kích cho camera. Camera chụp ảnh với thời gian màn trập thay đổi tùy theo điều kiện ánh sáng và gửi giá trị này cho máy tính nhúng. Máy tính nhúng tính toán độ lệch pha của xung kích camera dựa trên thời gian màn trập hiện tại và các độ trễ cố định khác trong hệ thống, rồi gửi cho vi điều khiển thực thi.



Hình 5.9. Mô hình phần cứng hệ thống StereoVision/INS/GPS

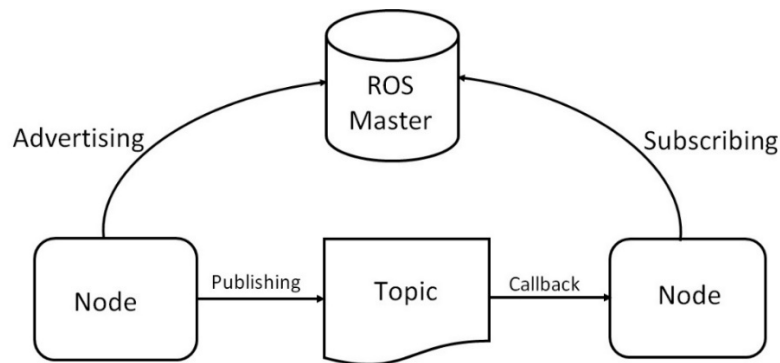
Hệ thống StereoVision/INS/GPS được lắp đặt như Hình 5.9. Stereo camera và IMU được gắn chặt vào nhau để tránh làm sai lệch kết quả hiệu chuẩn. Tất cả thiết bị được đặt trong một không gian nhỏ gọn 18 x 16.5 x 12cm, phù hợp để triển khai trên robot di động.

5.1.2. Phần mềm

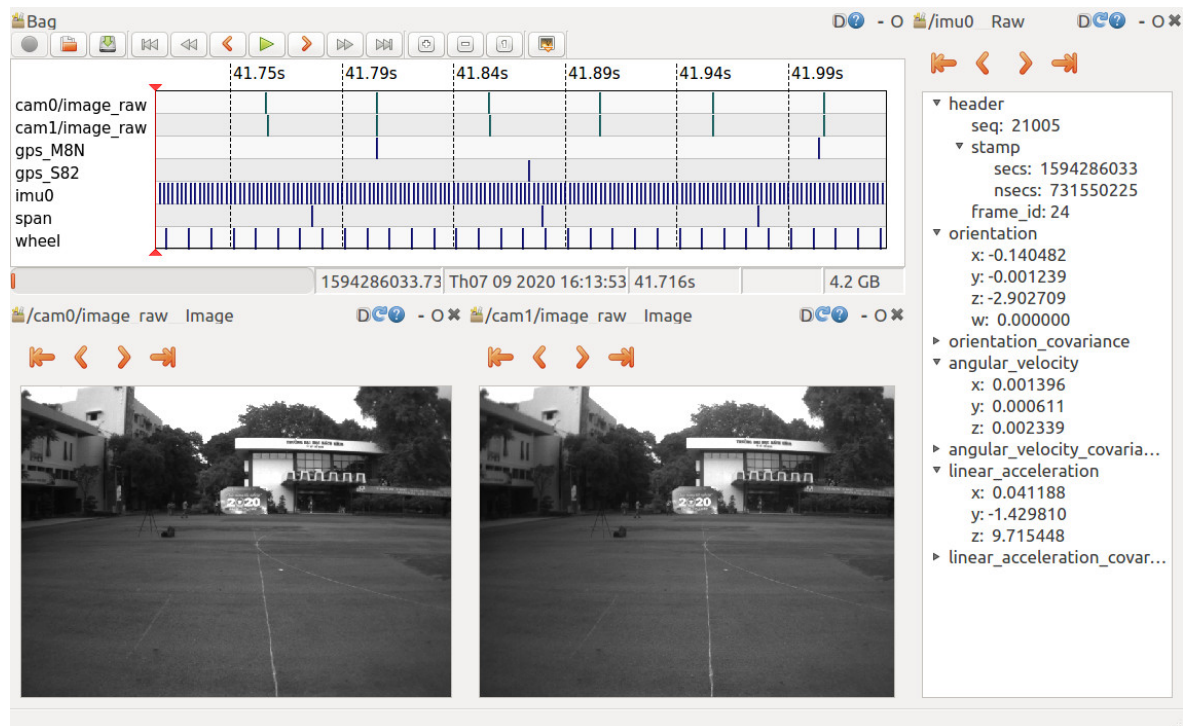
Phần mềm định vị tích hợp hoạt động trên máy tính nhúng chạy hệ điều hành Linux (Ubuntu 16.04). Ngôn ngữ C++ được sử dụng để lập trình thuật toán xử lý trong thời gian thực. Để tăng tốc độ phát triển phần mềm, đề tài chọn khai thác nền tảng ROS (Robot Operating System) [31].

ROS là nền tảng mã nguồn mở, bao gồm các bộ thư viện và công cụ trợ giúp cho việc phát triển ứng dụng robot. ROS được thiết kế theo triết lý phân tán, trong đó mỗi nút (node) là một quá trình xử lý độc lập. Các node trong ROS truyền nhận dữ liệu

với nhau dưới dạng bản tin (message) thông qua chủ đề (topic). Theo đó, node truyền sẽ quảng bá (advertise) chủ đề cần truyền với đơn vị kiểm soát là rosmaster. Node nhận sẽ đăng kí (subscribe) với rosmaster chủ đề cần nhận. Khi có cả node truyền và node nhận ở cùng một chủ đề, rosmaster sẽ thiết lập kết nối giữa hai bên để giao tiếp xảy ra, như Hình 5.10. Một chủ đề chỉ có một node phát, trong khi có thể có nhiều node nhận. Cơ chế này giúp các node giao tiếp một cách linh hoạt, chỉ cần biết chủ đề mà không cần biết cụ thể node ở đâu còn lại. Hơn nữa, rosmaster còn có thể kết nối các node trên các máy tính khác nhau trong cùng một mạng Ethernet, giúp cho việc truyền nhận dữ liệu giữa các thiết bị trở nên đơn giản như trên cùng một thiết bị.

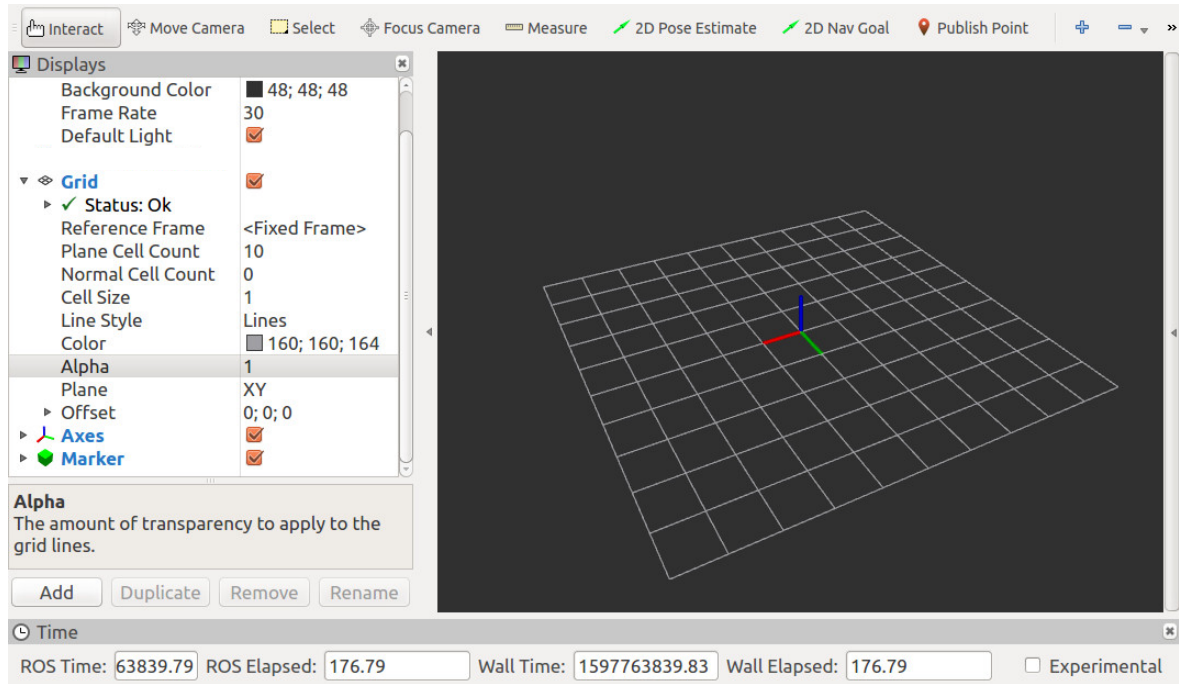


Hình 5.10. Cơ chế truyền nhận bản tin trong ROS



Hình 5.11. File dữ liệu bag

Công cụ rosbag cho phép lưu và phát lại các bản tin trong ROS dưới định dạng file bag. Bản tin được phát lại từ file bag có tính chất tương tự như lúc thu về mặt thời gian cũng như dữ liệu. File bag trong đề tài có dạng như Hình 5.11, trong đó thu lại dữ liệu đầu vào từ các thiết bị khi thử nghiệm và phát lại khi phát triển thuật toán.

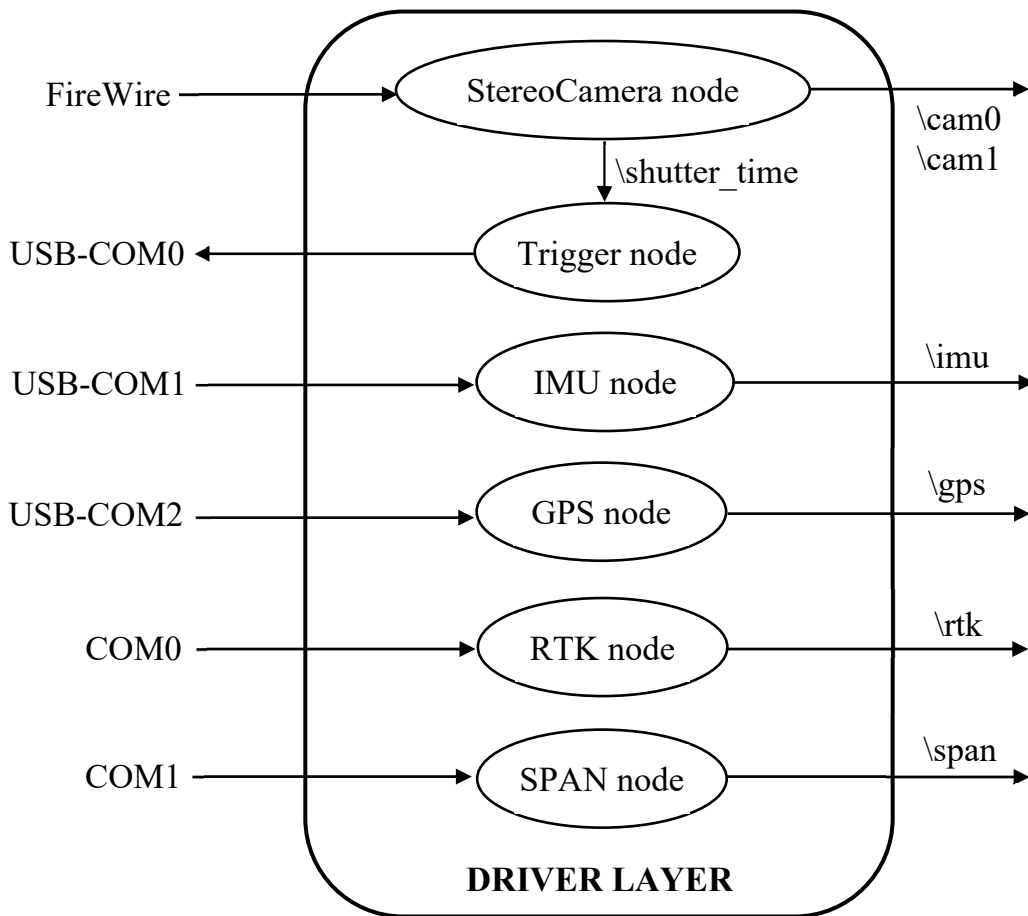


Hình 5.12. Môi trường ảo rviz

ROS cũng cung cấp môi trường ảo rviz cho phép hiển thị tọa độ 3D của vật thể dưới nhiều dạng khác nhau như Hình 5.12. Trong đề tài, rviz được dùng để hiển thị các quỹ đạo và landmark ước lượng một cách trực quan trên laptop người dùng.

Cấu trúc phần mềm hệ thống được chia thành ba lớp Driver, Process và Display. Trong đó, lớp Driver thực hiện giao tiếp với các thiết bị, chuyển đổi dữ liệu đọc về sang dạng bản tin ROS, kí hiệu là ‘\...’ như Hình 5.13. StereoCamera node đọc dữ liệu camera và chuyển sang ảnh xám 8-bit, đồng thời đọc thời gian màn trập hiện tại. Trigger node nhận thời gian màn trập, tính toán độ trễ xung kích và gửi ra vi điều khiển quản lý việc đồng bộ IMU-camera. IMU node đổi đơn vị của các đại lượng đọc từ IMU sang đơn vị chuẩn. GPS node và RTK node nhận bản tin dạng NMEA từ các thiết bị tương ứng, đọc tọa độ kinh-vĩ-cao độ và sai số của phép đo, chuyển đổi tọa

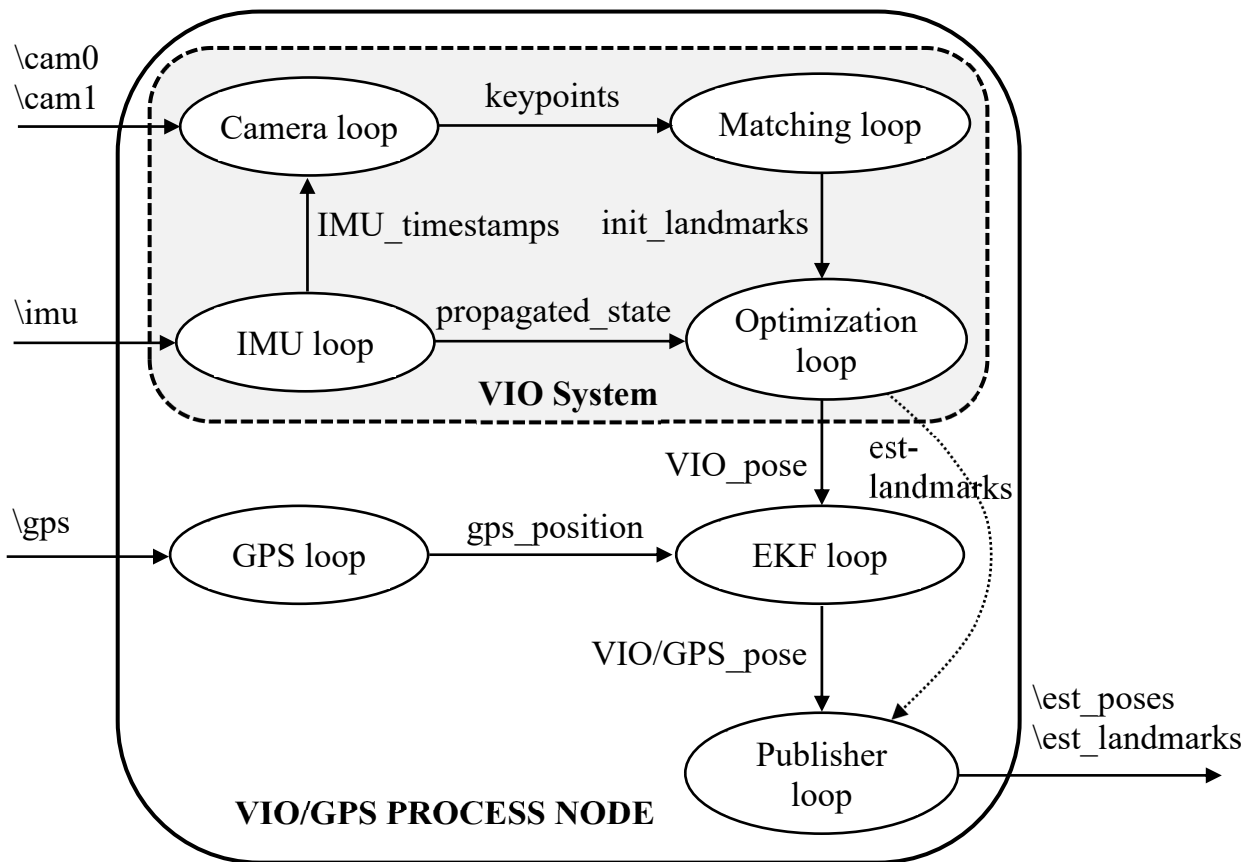
độ sang hệ ENU. SPAN node cũng thực hiện công việc tương tự như GPS node, nhưng trên đầu vào bản tin dạng NovAtel.



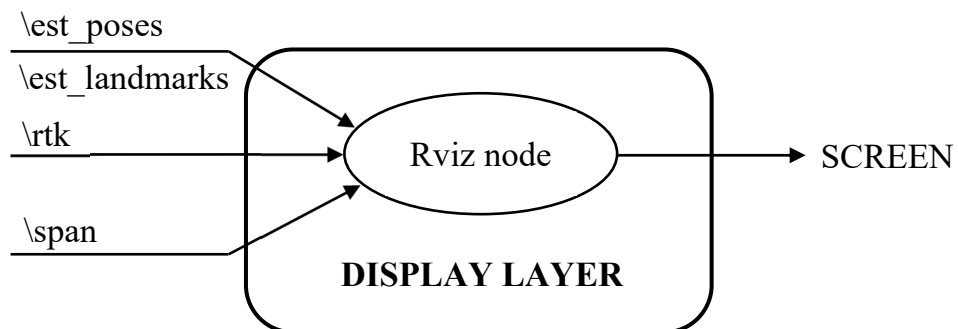
Hình 5.13. Cấu trúc phần mềm lớp Driver

Cấu trúc lớp Process, xử lý thuật toán tích hợp được trình bày trong Hình 5.14. Vì lớp này cần tốc độ thực thi nhanh, các luồng xử lý được lập trình song song và đặt chung trong một node, mỗi luồng chứa một vòng lặp vô cùng. Vòng lặp IMU thực hiện nhận phép đo IMU và lan truyền trạng thái hệ thống. Vòng lặp Camera nhận ảnh từ stereo camera và chạy thuật toán phát hiện điểm đặc trưng trong ảnh, đồng thời thực hiện ghép đồng bộ frame ảnh với các phép đo IMU theo dấu thời gian. Vòng lặp Matching nhận tập điểm đặc trưng, thực hiện ghép cặp giữa hai ảnh stereo và các ảnh trong quá khứ để khởi tạo các landmark 3D. Trong các thử nghiệm, thuật toán duy trì số lượng ảnh quá khứ là 5-7 cặp ảnh keyframe và 3-4 cặp ảnh gần nhất. Vòng lặp Optimization thực hiện tối ưu hóa tư thế robot và vị trí landmark từ trạng thái lan truyền và landmark khởi tạo. Vòng lặp GPS đọc vào các bản tin GPS. Vòng lặp EKF

ước lượng hiệu chỉnh tư thế robot từ hệ VIO và dữ liệu GPS. Cuối cùng, vòng lặp Publisher xuất ra tư thế ước lượng tích hợp và các landmark theo dạng bản tin ROS.



Hình 5.14. Cấu trúc phần mềm lớp Process



Hình 5.15. Cấu trúc phần mềm lớp Display

Sau khi lớp Driver và lớp Process xử lý thuật toán trên máy tính nhúng, lớp Display chạy trên laptop người dùng sẽ hiển thị kết quả ra màn hình thông qua môi trường ảo 3D rviz. Dữ liệu hiển thị gồm quỹ đạo và landmark ước lượng từ hệ VIO/GPS, cũng như các quỹ đạo chuẩn từ RTK và SPAN.

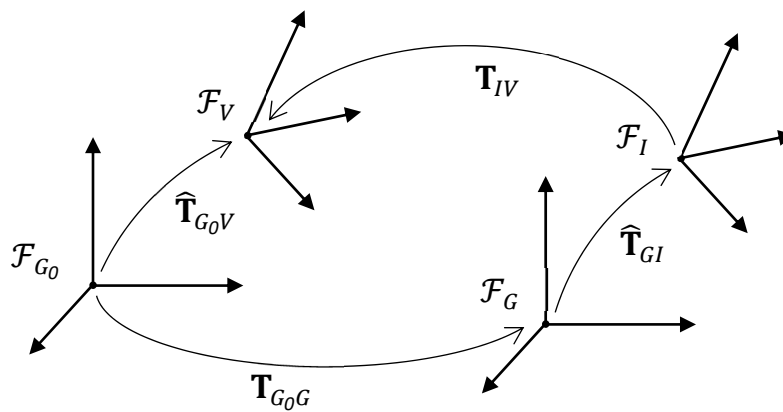
5.2. Phương pháp đánh giá sai số ước lượng

5.2.1. Quy đổi tọa độ

Do bố trí trong không gian, tồn tại khoảng cách giữa tâm của bộ ước lượng cần đánh giá (VIO/GPS) và tâm của bộ định vị chuẩn (RTK/SPAN). Bên cạnh đó, góc tọa độ của bộ ước lượng và bộ định vị chuẩn cũng có sai lệch thực tế khoảng vài mét do khác biệt trong thông số hiệu chỉnh bộ thu GPS của mỗi hãng. Vì vậy, để tính sai số ước lượng một cách chính xác, ta cần quy đổi tọa độ ngõ ra của bộ ước lượng sao cho cùng quan hệ tọa độ với bộ chuẩn. Xét ngõ ra bộ ước lượng VIO/GPS $\hat{\mathbf{T}}_{GI}$ và ngõ ra bộ chuẩn RTK/SPAN $\mathbf{T}_{G_0V}$ ($\mathcal{F}_{G_0}$ là góc tọa độ và $\mathcal{F}_V$ là tâm bộ chuẩn), tư thế ước lượng quy đổi sang bộ chuẩn là:

$$\hat{\mathbf{T}}_{G_0V} = \mathbf{T}_{G_0G} \hat{\mathbf{T}}_{GI} \mathbf{T}_{IV} = \begin{bmatrix} \mathbf{R}_{G_0G} & \mathbf{t}_{G_0G} \\ 0 & 1 \end{bmatrix} \hat{\mathbf{T}}_{GI} \begin{bmatrix} \mathbf{R}_{IV} & \mathbf{t}_{IV} \\ 0 & 1 \end{bmatrix} \quad (5.1)$$

trong đó phép xoay $\mathbf{R}_{G_0G} = \mathbf{I}_3$ do $\mathcal{F}_{G_0}$ và $\mathcal{F}_G$ là hệ tọa độ ENU, phép xoay $\mathbf{R}_{IV} \approx \mathbf{I}_3$ do ta sắp đặt hướng của bộ ước lượng và bộ chuẩn gần như trùng nhau trong thực nghiệm, khoảng cách $\mathbf{t}_{G_0G}$ được xác định bằng thí nghiệm tính chênh lệch vị trí trung bình giữa hai bộ định vị khi đặt đứng yên tại cùng một điểm trên mặt đất trong một khoảng thời gian (≥ 10 phút), và khoảng cách $\mathbf{t}_{IV}$ được đo bằng thước milimét. Phép quy đổi tọa độ được minh họa trong Hình 5.16.



Hình 5.16. Quy đổi tọa độ giữa bộ ước lượng và bộ chuẩn

Sau khi đã thống nhất quan hệ tọa độ giữa bộ ước lượng và bộ chuẩn, ta tiến hành đánh giá sai số giữa $\hat{\mathbf{T}}_{G_0V}$ và $\mathbf{T}_{G_0V}$ theo hai tiêu chuẩn: sai số tuyệt đối và sai số tương

đối. Để ngắn gọn và mang tính tổng quát hơn, chú thích G_0V được lược đi trong các quy ước sai số dưới đây.

5.2.2. Sai số tuyệt đối

Sai số tuyệt đối là sai lệch giữa tọa độ ước lượng $(\hat{\mathbf{p}}, \hat{\mathbf{R}})$ và tọa độ đúng $(\mathbf{p}, \mathbf{R})$ trong hệ tọa độ tuyệt đối ($\mathcal{F}_{G_0}$):

$$\begin{cases} \Delta^a \mathbf{p} = \hat{\mathbf{p}} - \mathbf{p} \\ \Delta^a \mathbf{R} = \hat{\mathbf{R}} \mathbf{R}^T \end{cases} \quad (5.2)$$

Sai số hướng xoay theo ba góc Euler: roll $\Delta^a \varphi$, pitch $\Delta^a \theta$, yaw $\Delta^a \psi$ được suy ra từ $\Delta^a \mathbf{R}$ theo công thức (4.23), (4.24) và (4.25).

Bên cạnh sai số theo từng trục, ta có thể tổng hợp sai số theo mỗi 3 trục vị trí và góc xoay. Sai số vị trí 3D được tính bằng khoảng cách Euclide giữa điểm ước lượng và điểm đúng:

$$\Delta^a d = \|\hat{\mathbf{p}} - \mathbf{p}\| = \sqrt{\Delta^a \mathbf{p}^T \Delta^a \mathbf{p}} \quad (5.3)$$

Sai số góc xoay 3D là độ lớn của góc xoay ở dạng biểu diễn trục-góc (axis-angle) của phép xoay $\Delta^a \mathbf{R}$:

$$|\Delta^a \alpha| = \arccos \left(\frac{\text{tr}(\Delta^a \mathbf{R}) - 1}{2} \right) \quad (5.4)$$

Để đánh giá sai số ước lượng trên cả quỹ đạo, ta dùng tiêu chuẩn sai số RMS (Root Mean Square Error), gọi x là đại lượng cần đánh giá:

$$\text{RMSE}(x) = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta^a x_i)^2} \quad (5.5)$$

5.2.3. Sai số tương đối

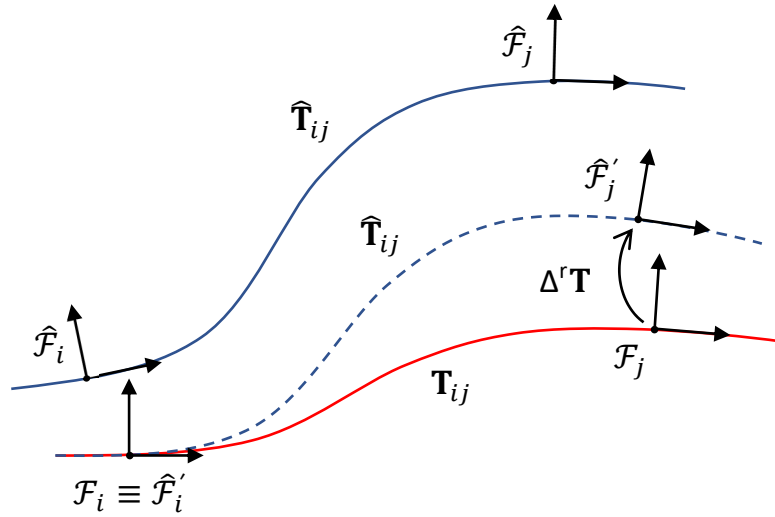
Khái niệm sai số tương đối được đề xuất trong bài báo [32] và được sử dụng rộng rãi trong các nghiên cứu về định vị thị giác-quán tính [8] [16] [33]. Sai số tương đối được định nghĩa là sai lệch giữa lượng thay đổi tương đối từ thời điểm i đến j của tọa độ ước lượng $\hat{\mathbf{T}}_{ij}$ và tọa độ đúng $\mathbf{T}_{ij}$:

$$\Delta^r \mathbf{T} = \mathbf{T}_{ij}^{-1} \hat{\mathbf{T}}_{ij} \quad (5.6)$$

trong đó:

$$\begin{aligned} \hat{\mathbf{T}}_{ij} &= \hat{\mathbf{T}}_i^{-1} \hat{\mathbf{T}}_j \\ \mathbf{T}_{ij} &= \mathbf{T}_i^{-1} \mathbf{T}_j \end{aligned} \quad (5.7)$$

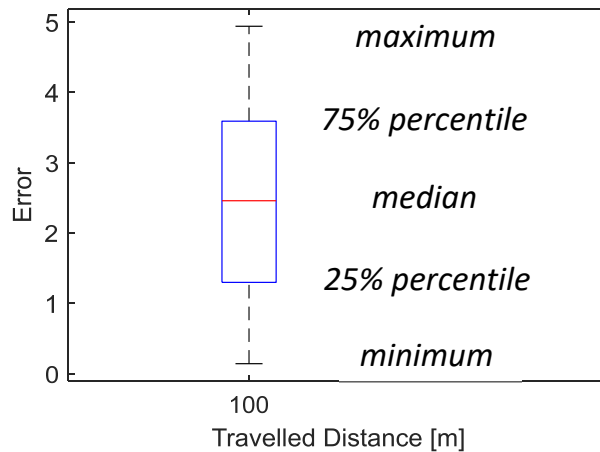
Có thể hiểu cách tìm sai số tương đối như minh họa ở Hình 5.17. Trước tiên ta căn chỉnh quỹ đạo ước lượng (đường màu xanh) với quỹ đạo đúng (đường màu đỏ) tại thời điểm i , sau đó tính sai lệch giữa quỹ đạo ước lượng đã căn chỉnh với quỹ đạo đúng tại thời điểm j . Hình 5.17.



Hình 5.17. Sai số tương đối giữa quỹ đạo ước lượng và quỹ đạo đúng

Khái niệm sai số tương đối cho phép ta tính và thống kê sai số ước lượng đối với một khoảng cách di chuyển nhất định từ những điểm bắt đầu khác nhau, nhờ đó bao quát được chất lượng định vị trên cả quỹ đạo.

Trong các kết quả dưới đây, ứng với mỗi độ dài quãng đường, học viên chọn ra $N = 100$ điểm bắt đầu cách đều nhau trên quỹ đạo, từ đó lập ra đồ thị thống kê dạng *boxplot* thể hiện phân bố của sai số theo 5 mức: nhỏ nhất, lớn nhất, trung vị, 25% và 75% số lượng như Hình 5.18. Đơn vị sai số vị trí là m hoặc % (m/m) khi chia cho quãng đường di chuyển, tương ứng đơn vị sai số góc xoay là ° hoặc °/m.



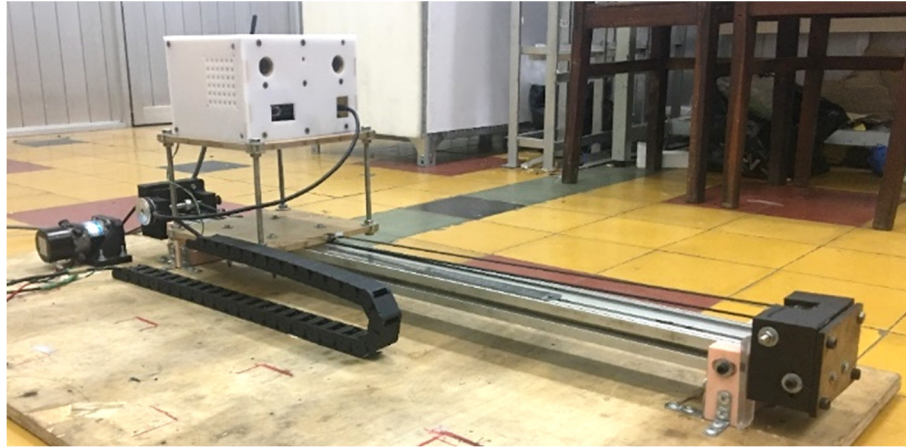
Hình 5.18. Dạng đồ thị boxplot

5.3. Kết quả thử nghiệm trong nhà

Trong phần này, học viên thực hiện đánh giá giải thuật VIO ở môi trường trong nhà, không có GPS thông qua hai thử nghiệm chuyển động tịnh tiến với thanh trượt và đi bộ nhiều vòng trên mặt phẳng.

5.3.1. Thử nghiệm với thanh trượt tuyến tính

Nhằm kiểm tra tính ổn định của giải thuật VIO với chuyển động tịnh tiến, hệ định vị được gắn lên thanh trượt tuyến tính như Hình 5.19. Thanh trượt được kéo tiến lùi liên tục bởi một động cơ DC servo. Chuyển động tạo ra có dạng sóng sin theo thời gian với tần số 0.3Hz, biên độ đỉnh-đỉnh 0.6m và vận tốc tối đa 0.56m/s. Vị trí chuẩn của thanh trượt được xác định bởi encoder với độ phân giải quy đổi là 3.3 μ m/xung encoder. Thanh trượt được đặt trong nhà dưới ánh sáng đèn huỳnh quang chớp tắt theo tần số điện 50Hz. Để chứng tỏ giải thuật có khả năng chuyển động thứ yếu trong môi trường, học viên bố trí hai người hoạt động ngẫu nhiên trong tầm nhìn của stereo camera như Hình 5.20. Thử nghiệm kéo dài gần 6 phút với tổng quãng đường di chuyển 120m.

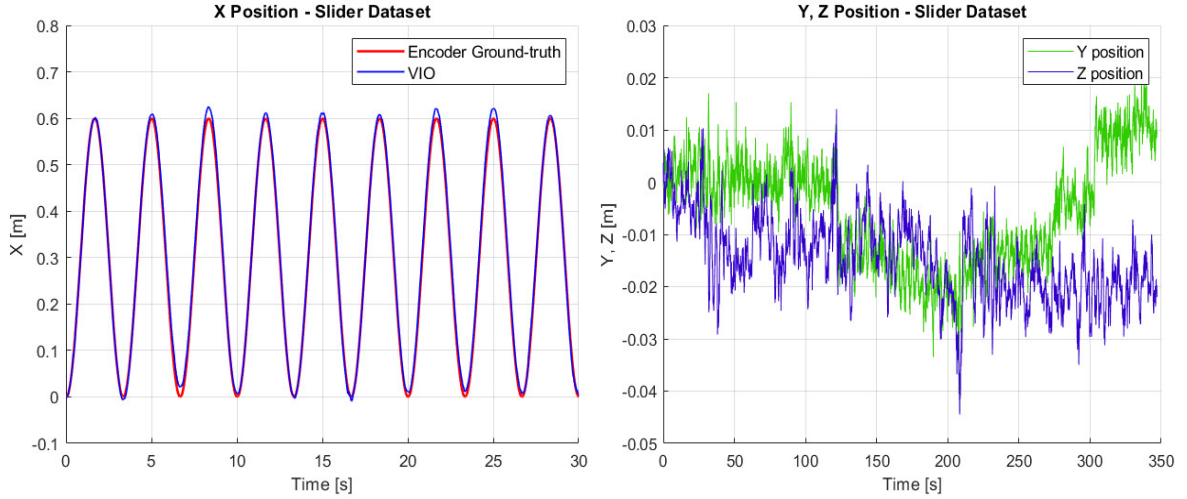


Hình 5.19. Thanh trượt dùng trong thử nghiệm

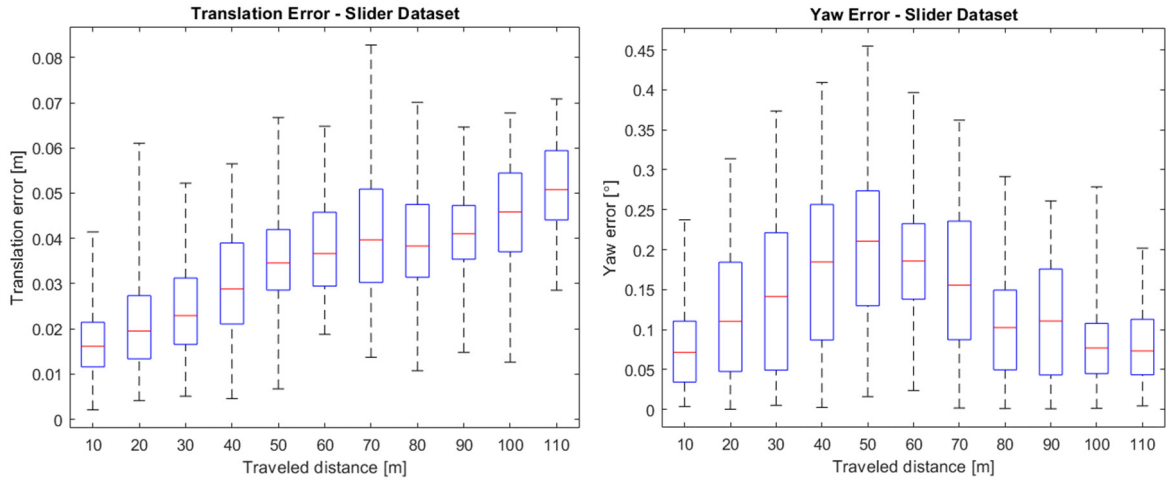


Hình 5.20. Ảnh trái và phải từ stereo camera trong tập dữ liệu Indoor-Slider

Vị trí ước lượng được thể hiện tại Hình 5.21, trong đó trục X của hệ VIO được căn chỉnh theo phương tới của thanh trượt để thu chuyển động hình sin, trong khi trục Y và Z chỉ chứa các rung lắc khi di chuyển. Vị trí ước lượng trục X bám sát quỹ đạo encoder, chỉ khác biệt nhỏ ở đỉnh sóng sin khi đổi chiều. Hình 5.22 cho thấy sai số ước lượng tăng chậm theo quãng đường. Sai số vị trí đạt trung bình 0.05 m sau 110 m di chuyển (0.045%), trong khi sai số góc yaw hầu như dưới 0.3 độ trên cả quãng đường. Độ chính xác cao đạt được không chỉ nhờ giải thuật ước lượng hoạt động tốt, mà còn do điều kiện thử nghiệm khá lý tưởng: môi trường giàu điểm đặc trưng, chuyển động tịnh tiến một trục trong phạm vi ngắn, giúp hệ VIO giữ được nhiều điểm đặc trưng cũ theo thời gian, nhờ đó hạn chế độ trôi trên các trục.



Hình 5.21. Vị trí ước lượng hệ VIO, tập dữ liệu Indoor-Slider



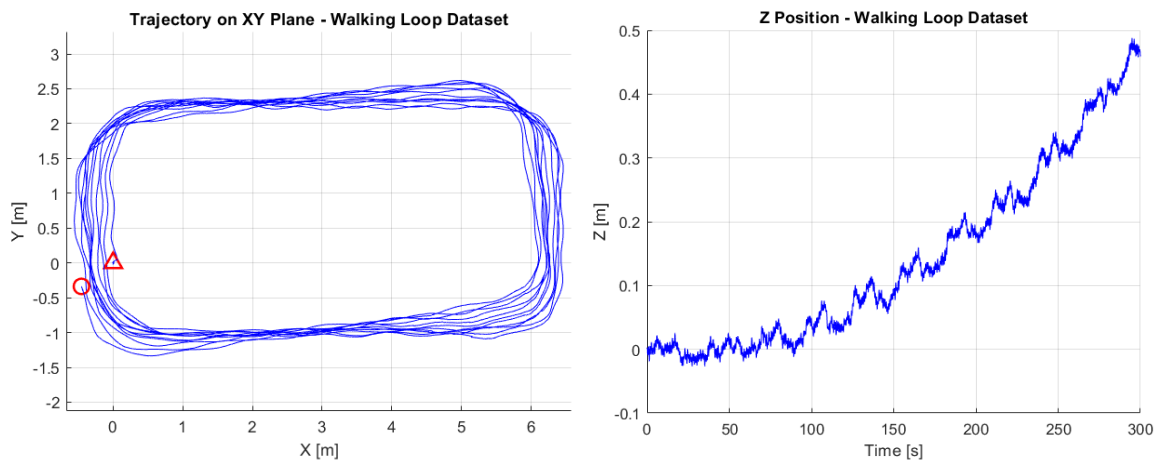
Hình 5.22. Sai số ước lượng vị trí và góc yaw, tập dữ liệu Indoor-Slider

5.3.2. Thử nghiệm với người đi bộ

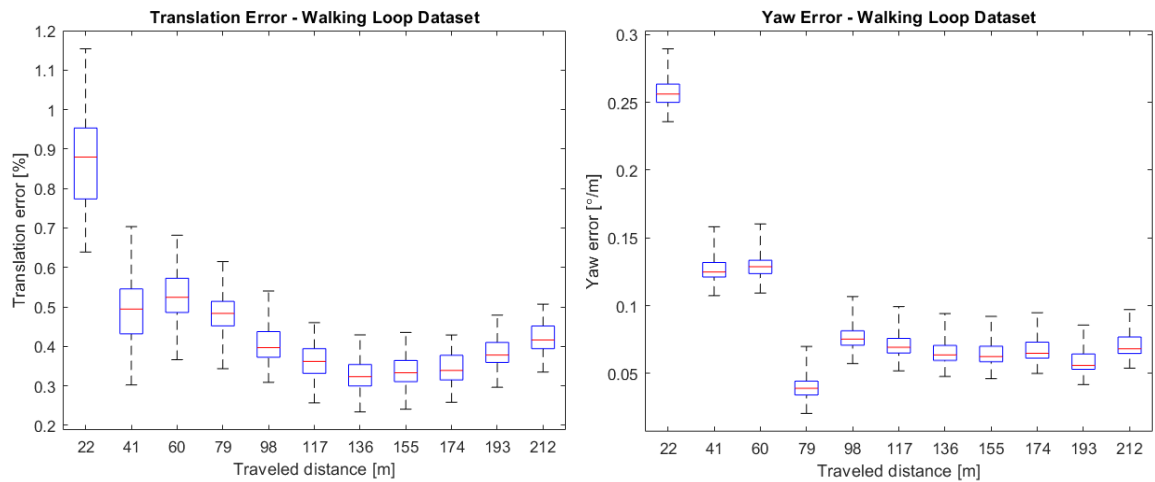
Để đánh giá giải thuật VIO trên chuyển động thực tế, học viên thử nghiệm cầm hệ thống trên tay và đi bộ nhiều vòng quanh phòng. Quỹ đạo di chuyển được kẻ trên mặt sàn là một hình chữ nhật kích thước 6x3 m với bán kính cong 0.6 m ở các góc. Tổng quãng đường dài 220 m, đi qua 11 vòng trong thời gian 5 phút, tốc độ trung bình là 0.73m/s. Do điều kiện thử nghiệm trong nhà không có thiết bị đo vị trí chuẩn, học viên chọn cách tính sai số mỗi khi hệ định vị quay lại điểm bắt đầu trên thực tế. Vì sai số lặp lại không liên tục theo thời gian, ta không thể áp dụng cách thống kê sai số theo N đoạn nhỏ như các thí nghiệm khác. Thay vào đó, thuật toán được chạy N lần trên cùng tập dữ liệu để tạo ra sai số thống kê tại mỗi điểm lặp. Về mặt hướng

xoay, kết quả ước lượng được so sánh với bộ ước lượng nội trên cùng IMU, có sai số RMS góc yaw động là 0.3° .

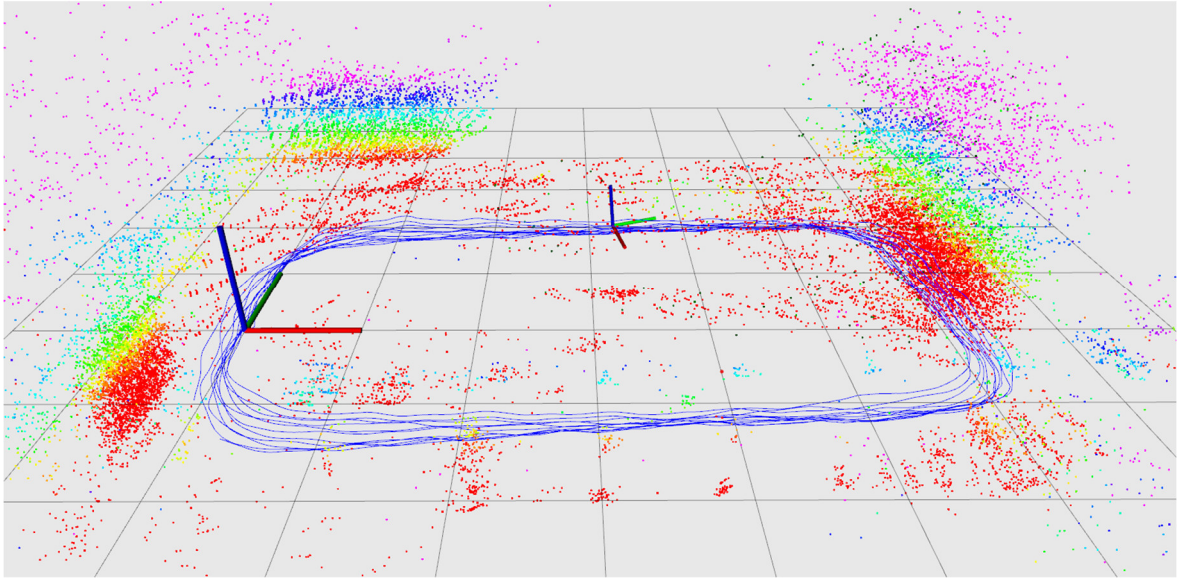
Hình 5.23 cho thấy quỹ đạo ước lượng bám sát quỹ đạo thực ở thời điểm ban đầu, nhưng sau đó có hiện tượng rời xa và lệch hướng do ảnh hưởng của sai số tích lũy. Dù vậy, sai số ước lượng bắt đầu từ vòng thứ 4 được duy trì dưới 0.5% quãng đường đối với vị trí 3D, và dưới $0.1^\circ/\text{m}$ đối với góc yaw, như Hình 5.24. Các landmark 3D trong quá trình định vị có thể được dùng để tái tạo hình dáng căn phòng như Hình 5.25.



Hình 5.23. Vị trí ước lượng VIO, tập dữ liệu Indoor-Walking



Hình 5.24. Sai số ước lượng vị trí và góc yaw, tập dữ liệu Indoor-Walking



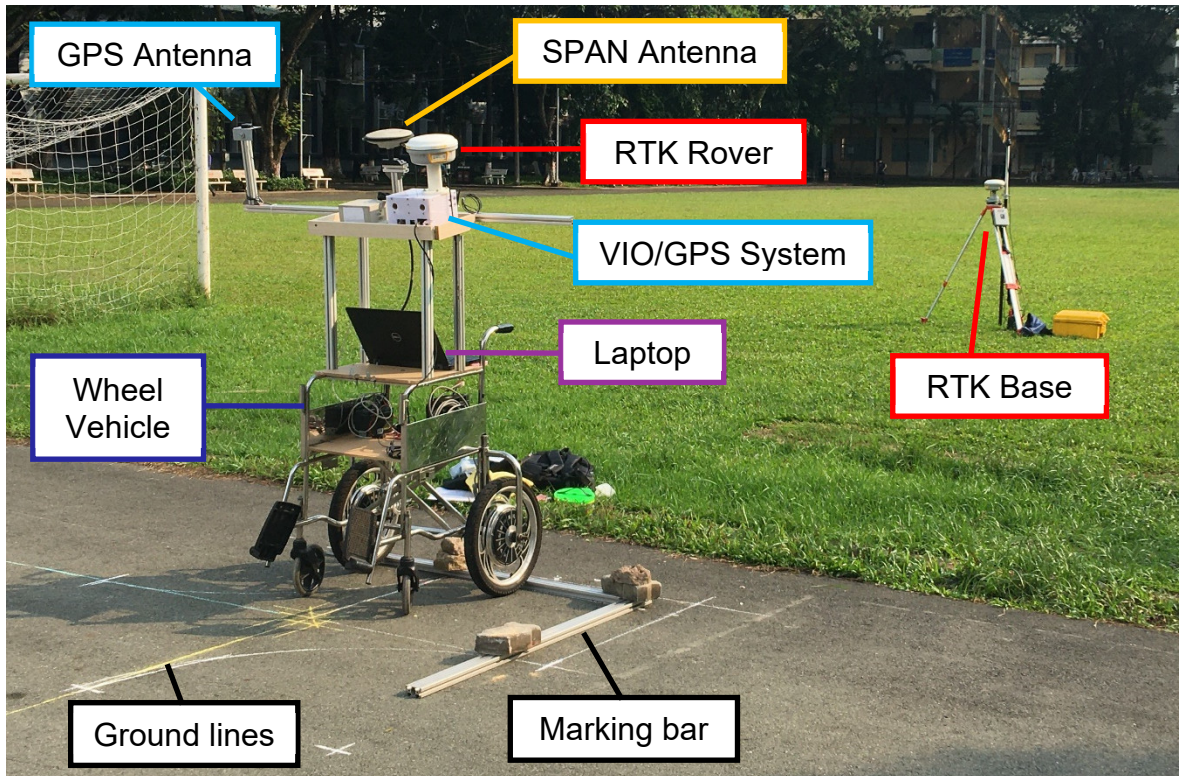
Hình 5.25. Ảnh tái dựng căn phòng từ các landmark thừa thớt và quỹ đạo ước lượng trong thử nghiệm Indoor-Walking

Hai thử nghiệm trong nhà đã chứng minh giải thuật VIO trong đề tài hoạt động đúng, ổn định và có mức trôi thấp. Để hạn chế độ trôi hơn nữa, ta có thể áp dụng phương pháp phát hiện vòng kín, trong đó quỹ đạo ước lượng được căn chỉnh lại mỗi khi bộ định vị hoàn tất một vòng [34]. Tuy nhiên, phương pháp này đòi hỏi hệ thống duy trì bản đồ toàn cục và ghép cặp một lượng lớn điểm đặc trưng trong quá khứ để tìm ra vòng kín, khiến cho khối lượng tính toán tăng mạnh. Do đó, ta cần cân nhắc khả năng xử lý của phần cứng trước khi thực hiện.

5.4. Kết quả thử nghiệm ngoài trời

Ở điều kiện ngoài trời, dữ liệu định vị tuyệt đối từ GPS có thể dùng để khử trôi kết quả VIO thông qua thuật toán tích hợp. Trong phần này, học viên tiến hành thử nghiệm hệ thống VIO/GPS khi di chuyển ngoài trời với nhiều dạng quỹ đạo và cự ly khác nhau. Địa điểm thử nghiệm bao gồm trong sân trường Đại học Bách Khoa Thành phố Hồ Chí Minh với môi trường tĩnh ít người qua lại; và ngoài đường phố với môi trường động xe chạy liên tục. Dữ liệu vị trí chuẩn được lấy từ hệ GPS-RTK với độ chính xác cấp centimét, đồng thời dữ liệu góc xoay chuẩn được lấy từ hệ SPAN với độ chính xác dưới 0.1 độ [30]. Để tiện di chuyển, các thiết bị được gắn cố định lên xe đẩy như Hình 5.26. Khoảng cách giữa hệ VIO và anten GPS trong các thử nghiệm là

từ 0.5 đến 1m, nhằm cập nhật tốt góc yaw theo công thức (4.37). Hệ VIO được đặt nghiêng một góc 10 độ về phía trước để giảm diện tích phần bầu trời trong ảnh chụp, tăng diện tích phần vật thể chứa nhiều điểm đặc trưng. Các đường kẻ, thanh chặn trên mặt đất đóng vai trò tăng tính lặp lại của chuyển động, cũng như cho biết kích thước thực sự của quỹ đạo. Hệ VIO/GPS giao tiếp với laptop qua WiFi để nhận lệnh và trả kết quả. Góc nhìn của stereo camera tại vị trí xuất phát được thể hiện ở Hình 5.27.



Hình 5.26. Bố trí thử nghiệm ngoài trời

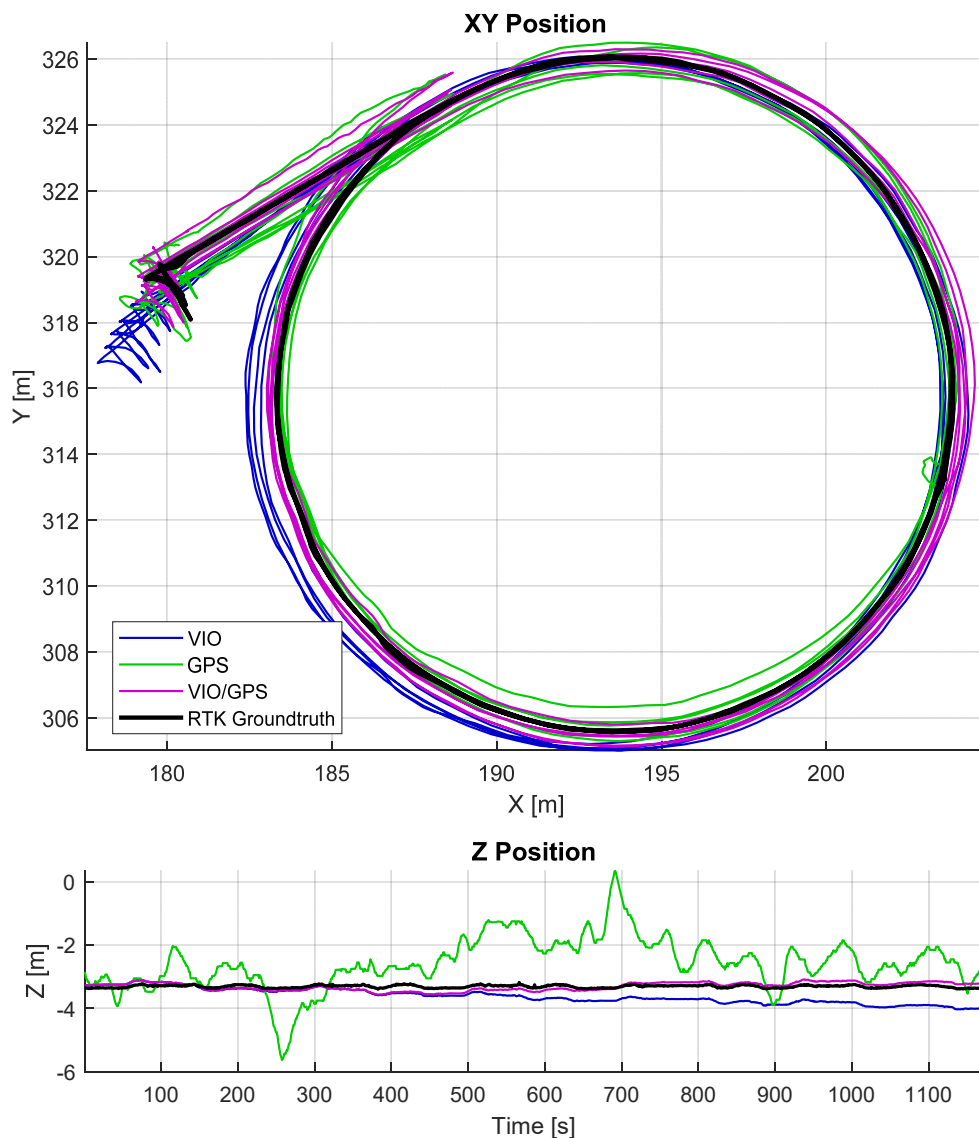


Hình 5.27. Ảnh stereo camera tại điểm xuất phát trong thử nghiệm trên sân trường

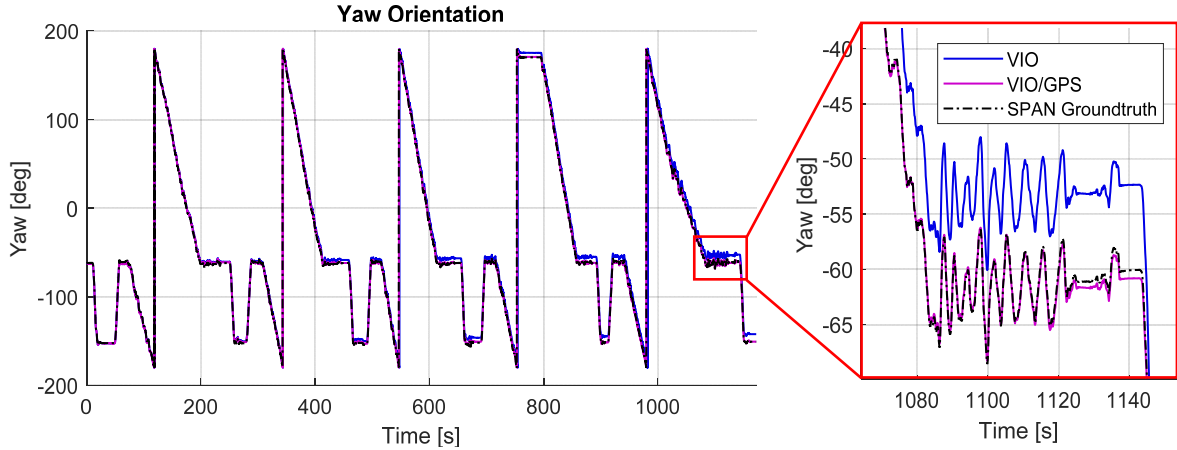
5.4.1. Thử nghiệm trên sân trường

5.4.1.1. Di chuyển theo hình tròn

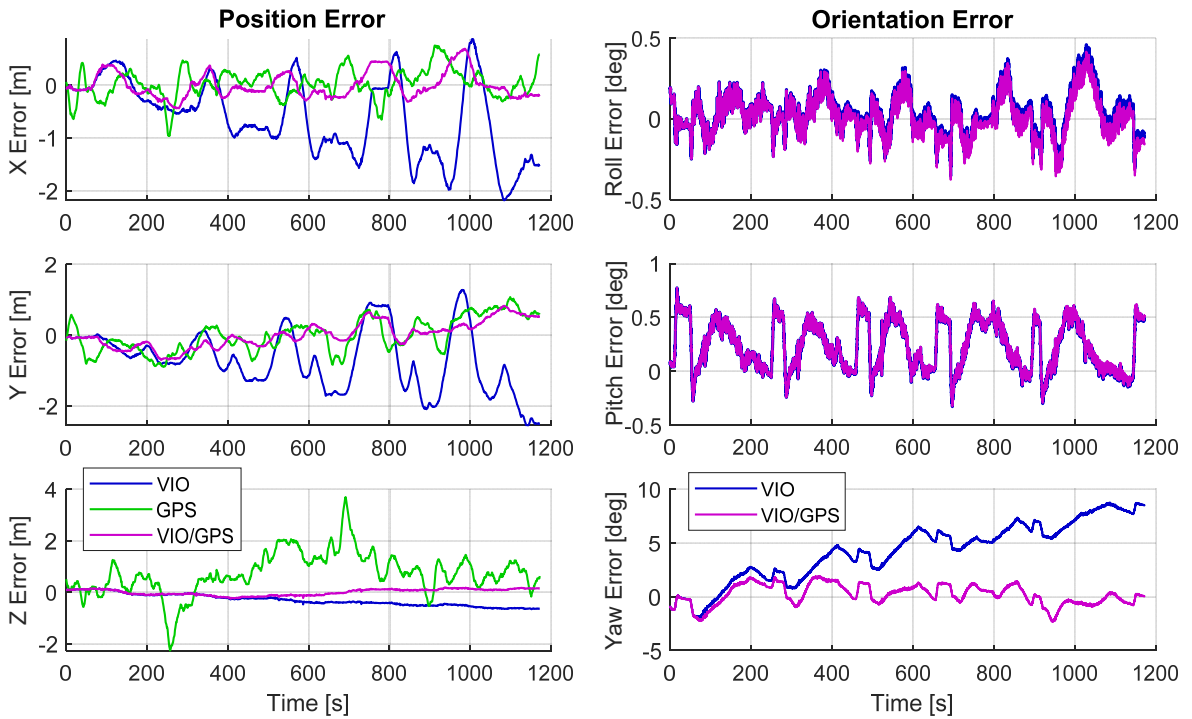
Trong thử nghiệm đầu tiên, xe đẩy di chuyển 5 vòng theo quỹ đạo hình tròn đường kính 20m. Tốc độ di chuyển trung bình 0.5m/s không kể những lúc dừng lại. Sau mỗi vòng, xe được kéo lùi theo đường thẳng về điểm xuất phát đánh dấu bởi thanh chặn, nhằm thể hiện sự khác biệt giữa quỹ đạo ước lượng và quỹ đạo chuẩn. Theo Hình 5.28, quỹ đạo VIO bị trôi dần theo thời gian, quỹ đạo GPS không trôi nhưng bị phân tán do nhiễu tần số thấp, nhất là trên trục Z. Trong khi đó, quỹ đạo tích hợp VIO/GPS khắc phục được độ trôi đồng thời hạn chế được nhiễu. Hình 5.29 cũng cho thấy góc yaw hệ tích hợp không trôi so với giá trị chuẩn như hệ VIO.



Hình 5.28. Kết quả định vị vị trí, tập dữ liệu Outdoor-Circle

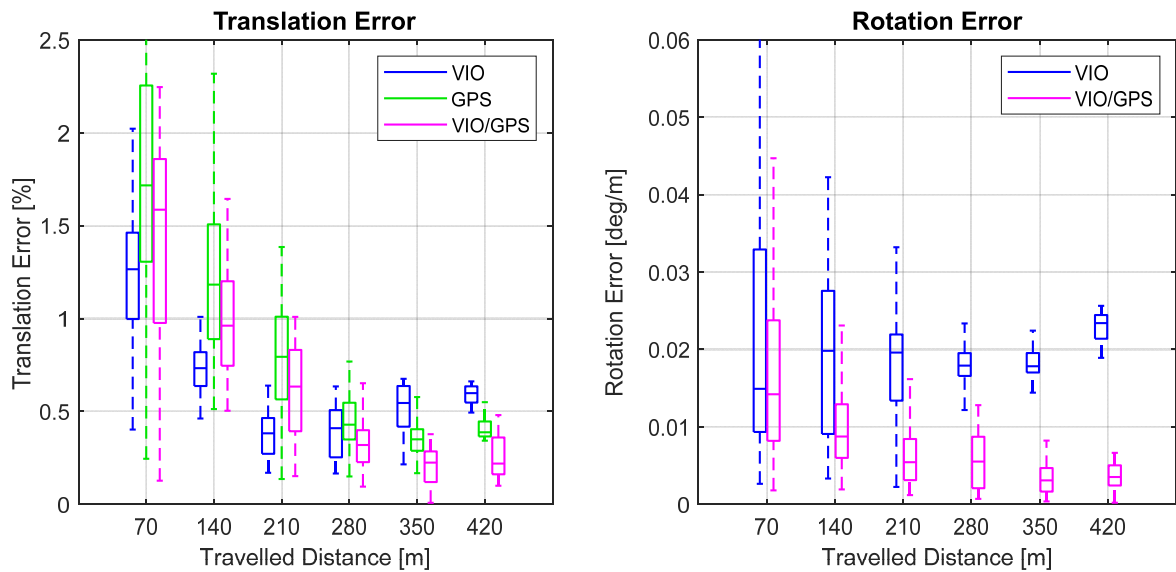


Hình 5.29. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Circle



Hình 5.30. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Circle

Theo Hình 5.30, sai số vị trí VIO có biên độ tăng dần trên cả 3 trục, trong khi biên độ sai số GPS và VIO/GPS hầu như không thay đổi. Đáng chú ý, sai số tích hợp trên trục Z gần như bằng 0. Sai số góc yaw hệ VIO bị trôi theo thời gian, trong khi với hệ tích hợp thì không thay đổi. Sai số góc roll và pitch của hai hệ luôn giữ trong một mức nhất định, cho thấy không cần ước lượng lại hai góc này trong hệ tích hợp.



Hình 5.31. Sai số tương đối vị trí và góc xoay theo quãng đường, tập dữ liệu Outdoor-Circle

Tập dữ liệu	Hệ định vị	Sai số RMS vị trí [m]			
		X	Y	Z	3D
Outdoor-Circle	VIO	0.9473	1.1077	0.3814	1.5066
	GPS	0.2732	0.4806	1.0885	1.2652
	VIO/GPS	0.2402	0.3582	0.1194	0.4475

Bảng 5.1. Sai số tuyệt đối vị trí, tập dữ liệu Outdoor-Circle

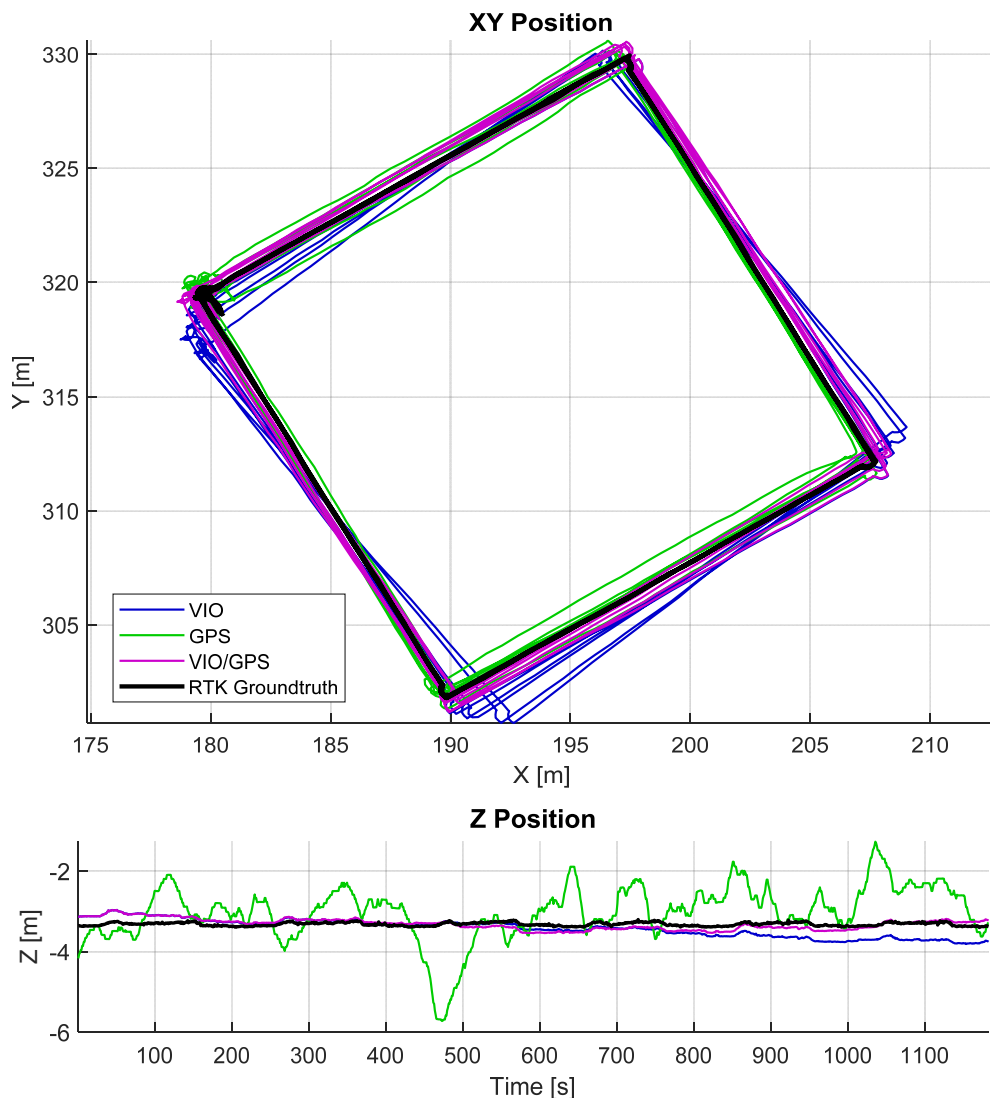
Tập dữ liệu	Hệ định vị	Sai số RMS góc xoay [deg]			
		φ	θ	ψ	3D
Outdoor-Circle	VIO	0.1236	0.3267	5.0591	5.0711
	VIO/GPS	0.1211	0.3289	0.9629	1.0247

Bảng 5.2. Sai số tuyệt đối góc xoay, tập dữ liệu Outdoor-Circle

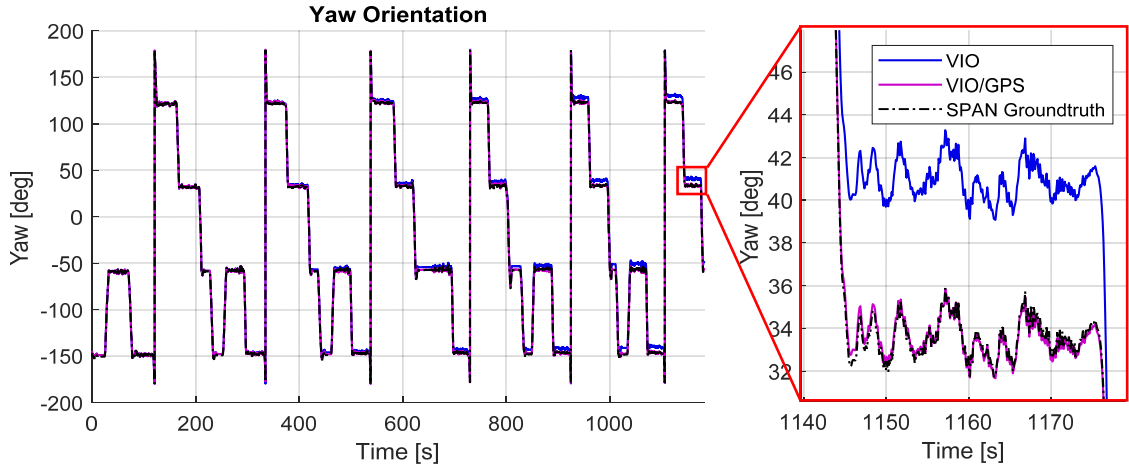
Thống kê sai số tương đối được trình bày trong Hình 5.31. Sau 420m di chuyển, sai số độ dời hệ VIO khoảng 0.5-0.7%, GPS khoảng 0.4-0.6% và VIO/GPS khoảng 0.2-0.5%. Sai số góc yaw VIO ổn định quanh mức $0.02^{\circ}/m$, trong khi góc yaw VIO/GPS đạt đến sai số $0.005^{\circ}/m$ ở cuối quãng đường, chứng tỏ giải thuật tích hợp có khả năng khử trôi góc yaw tốt. Bảng 5.1 và Bảng 5.2 cũng cho thấy các sai số tuyệt đối vị trí và góc xoay của hệ tích hợp VIO/GPS là nhỏ hơn đáng kể so với từng hệ riêng lẻ. Sai số RMS vị trí 3D dưới 0.5m và sai số góc yaw dưới 1° .

5.4.1.2. Di chuyển theo hình vuông

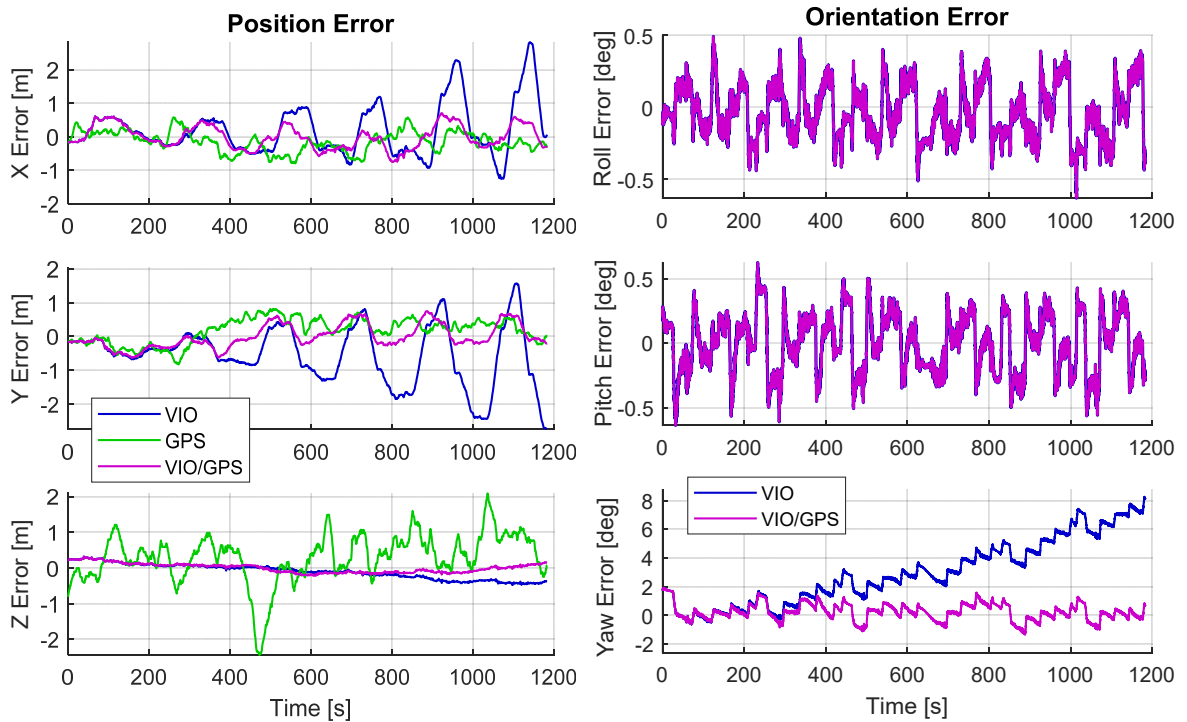
Trong thử nghiệm thứ 2, xe đẩy di chuyển 6 vòng theo quỹ đạo hình vuông kích thước 20x20m như Hình 5.32. Tổng quãng đường đi qua 515m với tốc độ trung bình 0.65m/s không kể các đoạn dừng sau mỗi vòng. Khác với quỹ đạo tròn, góc hướng trên quỹ đạo hình vuông thay đổi đột ngột hơn tại các khúc cua, như Hình 5.33. Khi xe xoay 90 độ, các điểm đặc trưng cũ trong khung hình nhanh chóng mất đi khiến cho phép đo thị giác nhất thời gặp khó khăn. Lúc này, phép đo vận tốc góc tốc độ cao trên IMU phát huy tác dụng bắt kịp độ thay đổi của góc hướng, giúp hệ VIO xử lý được chuyển động xoay.



Hình 5.32. Kết quả định vị vị trí, tập dữ liệu Outdoor-Square

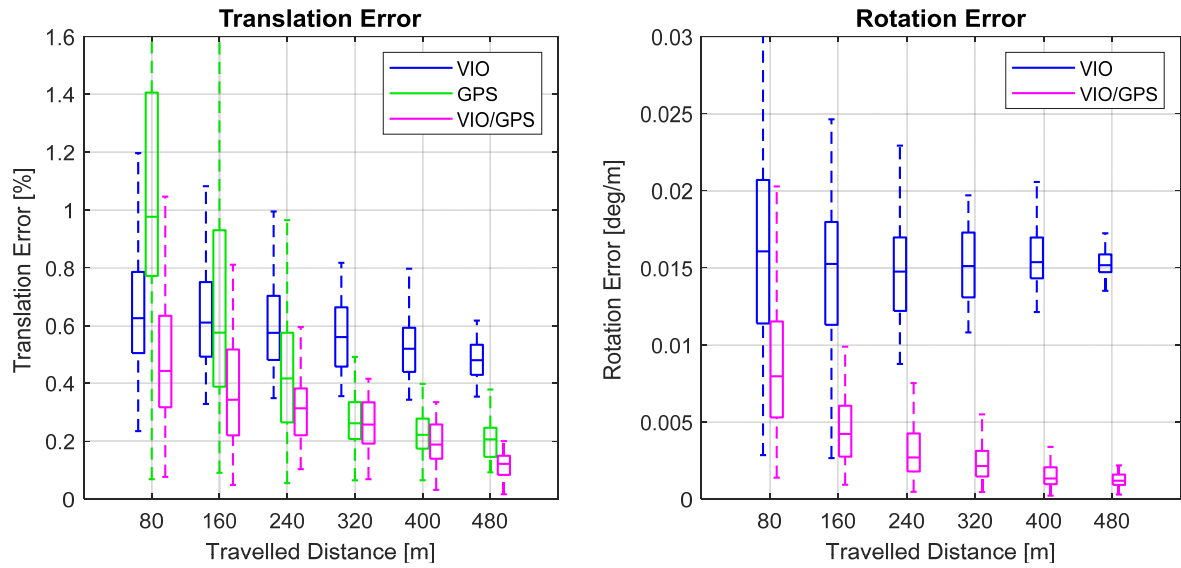


Hình 5.33. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Square



Hình 5.34. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Square

Hình 5.34 cho thấy sai số vị trí và góc yaw của hệ VIO ban đầu là nhỏ nhưng sau đó tăng dần theo thời gian do độ trôi. Sai số GPS không thay đổi nhưng dao động với biên độ lớn ở trục Z. Chỉ có hệ tích hợp VIO/GPS giải quyết được bài toán chính xác và không trôi. Sai số tương đối ở Hình 5.35 cho thấy hệ VIO có độ chính xác khoảng 0.5% đối với độ dời và 0.015°/m đối với góc xoay tổng hợp. Trong khi đó, phần trăm sai số của hệ VIO/GPS liên tục giảm theo quãng đường, còn hơn 0.1% với độ dời và 0.001°/m với góc xoay khi đi đến 480m.



Hình 5.35. Thống kê sai số vị trí và góc xoay tương đối theo quãng đường, tập dữ liệu Outdoor-Square

Tập dữ liệu	Hệ định vị	Sai số RMS vị trí [m]			
		X	Y	Z	3D
Outdoor-Square	VIO	0.7887	0.9997	0.2310	1.2942
	GPS	0.3310	0.4054	0.7440	0.9096
	VIO/GPS	0.3421	0.3444	0.1293	0.5024

Bảng 5.3. Sai số vị trí tuyệt đối, tập dữ liệu Outdoor-Square

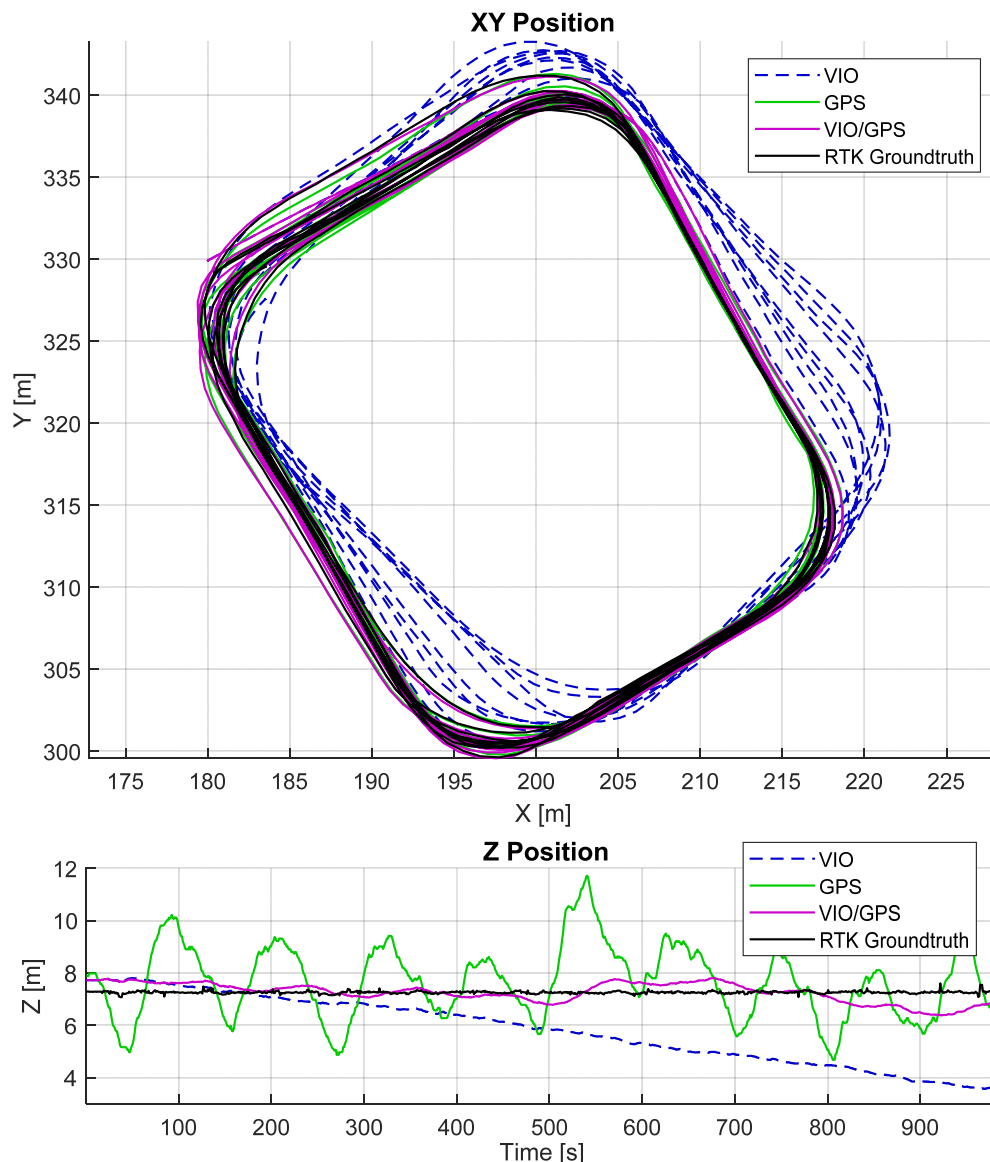
Tập dữ liệu	Hệ định vị	Sai số RMS góc xoay [deg]			
		φ	θ	ψ	3D
Outdoor-Square	VIO	0.1879	0.2253	3.7348	3.7463
	VIO/GPS	0.1869	0.2261	0.6358	0.7002

Bảng 5.4. Sai số góc xoay tuyệt đối, tập dữ liệu Outdoor-Square

Sai số vị trí tuyệt đối theo Bảng 5.3 tương đồng với thí nghiệm trước. Hệ VIO/GPS sai số thấp nhất trong ba hệ và xấp xỉ 0.5m đối với vị trí 3D. Trong khi đó, Bảng 5.4 cho thấy sai số góc yaw tốt hơn thí nghiệm trước, dưới 0.7° , một phần vì chuyển động chủ yếu trên đường thẳng, tuy có những thời điểm góc hướng thay đổi đột ngột nhưng độ trôi chung quy không lớn như trường hợp thay đổi liên tục trên quỹ đạo tròn.

5.4.1.3. Di chuyển theo hình chữ nhật bo góc

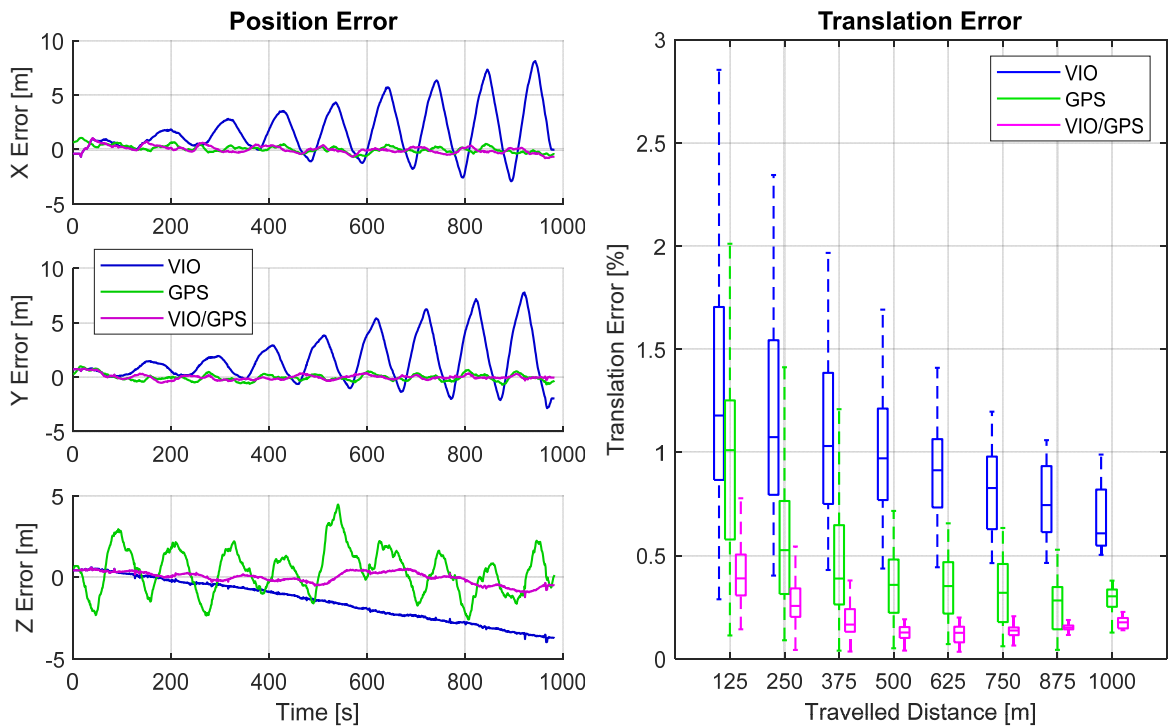
Trong thử nghiệm thứ 3, xe di chuyển 9 vòng theo hình chữ nhật bo góc 30x35m như Hình 5.36. Quãng đường dài hơn 1km với tốc độ trung bình 1m/s. Vị trí chuẩn được lấy từ hệ RTK, không có dữ liệu chuẩn cho góc xoay. Trên đường dài, ta thấy rõ ảnh hưởng của độ trôi lên hệ VIO, làm quỹ đạo ước lượng ngày càng cách xa quỹ đạo thực. Trong khi đó, hệ tích hợp VIO/GPS giữ được vị trí tuyệt đối theo thời gian, loại được độ trôi nhờ vào GPS.



Hình 5.36. Kết quả định vị vị trí, tập dữ liệu Outdoor-Rectangle

Hình 5.37 cho thấy sai số vị trí của hệ VIO/GPS ổn định ở mức nhỏ theo thời gian, trong khi sai số VIO ngày càng lớn. Tuy nhiên về mặt tương đối, hệ VIO đạt

được sai số 0.5-1% sau 1000m, trong khi đó hệ VIO/GPS có sai số xuống đến 0.2%. Theo Bảng 5.5, sai số RMS của hệ tích hợp VIO/GPS đạt 0.55m, không khác biệt nhiều so với hai thử nghiệm trước, mặc dù thử nghiệm này có quãng đường dài hơn và tốc độ di chuyển nhanh hơn gấp 2 lần. Điều đó cho thấy hệ định vị tích hợp có chất lượng tốt và ổn định trong nhiều điều kiện khác nhau.



Hình 5.37. Sai số vị trí theo thời gian (trái) và sai số vị trí tương đối theo quãng đường (phải), tập dữ liệu Outdoor-Rectangle

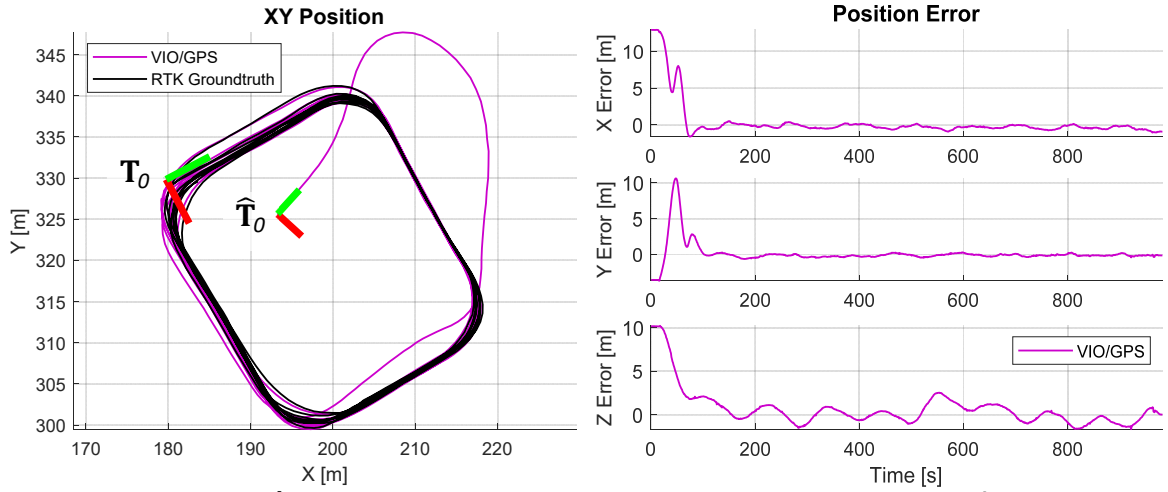
Tập dữ liệu	Hệ định vị	Sai số RMS vị trí [m]			
		X	Y	Z	3D
Outdoor-Rectangle	VIO	2.7480	2.6215	1.9307	4.2604
	GPS	0.3235	0.3239	1.4355	1.5067
	VIO/GPS	0.3171	0.2423	0.3745	0.5472

Bảng 5.5. Sai số RMS vị trí, tập dữ liệu Outdoor-Rectangle

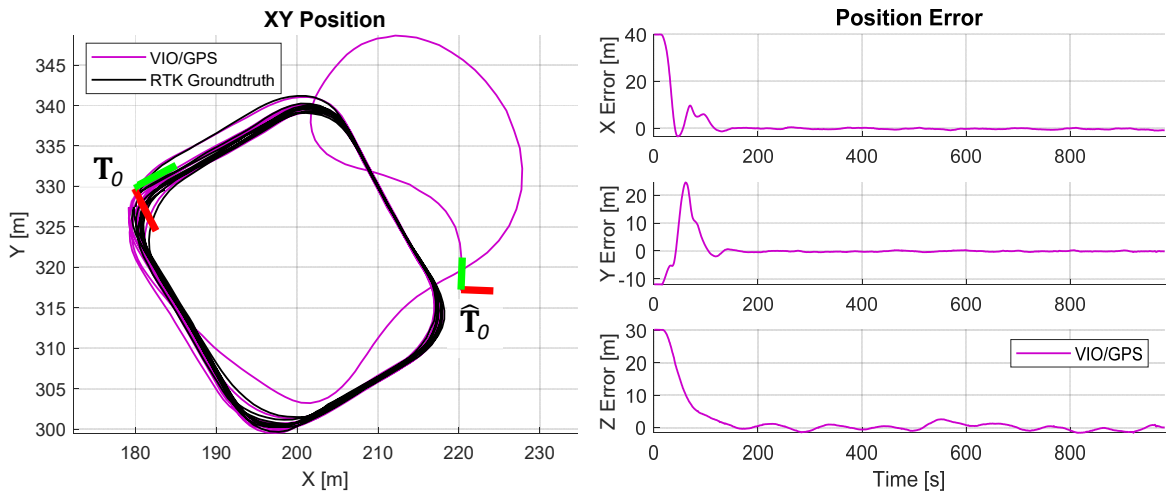
5.4.1.4. Thử nghiệm khả năng tự hiệu chỉnh

Trong thực tế, xảy ra trường hợp giá trị định vị bị sai lệch nhiều do mất tín hiệu GPS trong thời gian dài hoặc do bước căn chỉnh ban đầu không tốt. Thử nghiệm sau đây khảo sát khả năng tự hiệu chỉnh trở lại giá trị đúng của hệ tích hợp trong điều

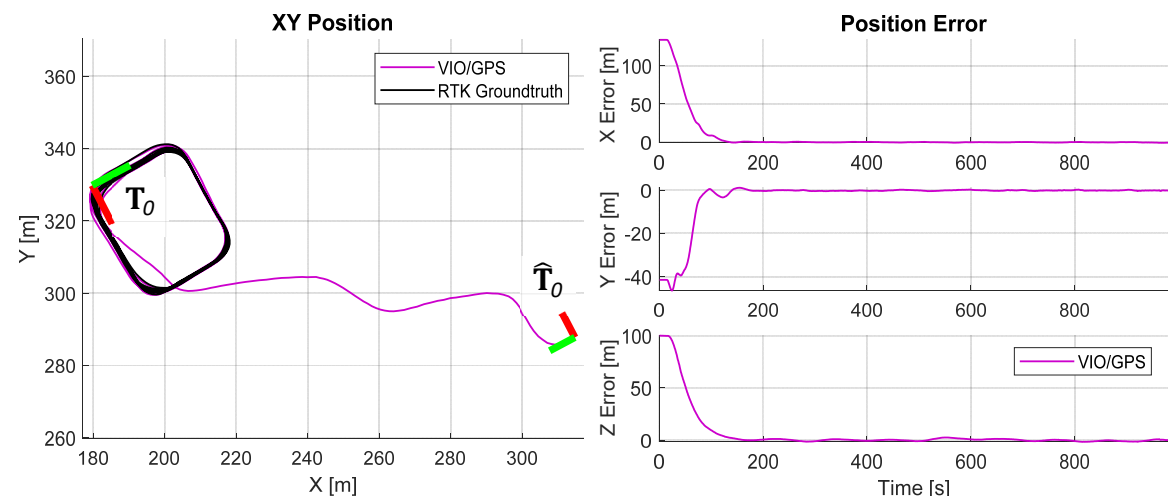
kiện có GPS. Tập dữ liệu sử dụng là *Outdoor-Rectangle* như thử nghiệm 3. Kết quả tự hiệu chỉnh trong ba trường hợp sai số ban đầu từ nhỏ đến lớn như sau:



Hình 5.38. Kết quả tự hiệu chỉnh vị trí 1, $\hat{x}_0 = \hat{y}_0 = \hat{z}_0 = 10\text{m}$, $\hat{\psi}_0 = 20^\circ$



Hình 5.39. Kết quả tự hiệu chỉnh vị trí 2, $\hat{x}_0 = \hat{y}_0 = \hat{z}_0 = 30\text{m}$, $\hat{\psi}_0 = 60^\circ$



Hình 5.40. Kết quả tự hiệu chỉnh vị trí 3, $\hat{x}_0 = \hat{y}_0 = \hat{z}_0 = 100\text{m}$, $\hat{\psi}_0 = 180^\circ$

Lần	Tọa độ ban đầu		Thời gian sai số giảm xuống dưới 2m [s]			
	$\hat{x}_0, \hat{y}_0, \hat{z}_0$ [m]	$\hat{\psi}_0$ [°]	X	Y	Z	TB
1	10, 10, 10	20	64.63	86.63	91.63	80.96
2	30, 30, 30	60	108.6	98.63	124.6	110.61
3	100, 100, 100	180	122.0	132.4	176.0	143.47

Bảng 5.6. Thời gian tự hiệu chỉnh của hệ VIO/GPS

Hình 5.38, Hình 5.39 và Hình 5.40 cho thấy quỹ đạo ước lượng từ vị trí khởi tạo sai được hiệu chỉnh dần theo thời gian trở lại giá trị đúng, sai số giảm về mức thông thường của hệ tích hợp. Theo Bảng 5.6, thời gian tự hiệu chỉnh tăng theo sai số ban đầu, nhưng tốc độ hội tụ cũng nhanh hơn nhờ vào hoạt động ước lượng tối ưu của bộ lọc EKF.

Qua các thử nghiệm trên, ta thấy thuật toán VIO và VIO/GPS hoạt động tốt ở môi trường ngoài trời với đặc trưng tĩnh. Sai số tương đối VIO trên dưới 1% và 0.02°/m, trong khi sai số tuyệt đối VIO/GPS trên dưới 0.5m và 1°. Hai hệ định vị có thể áp dụng cho robot mặt đất lẫn trên không nhờ ngõ ra đầy đủ 6 trục. Đặc biệt, hệ VIO/GPS có ngõ ra tọa độ toàn cục không trôi theo thời gian, thích hợp cho phương tiện làm việc trong thời gian dài mà không cần sử dụng thêm phương pháp hiệu chỉnh tọa độ nào khác. Hơn nữa, khi GPS không khả dụng, hệ tích hợp có khả năng duy trì kết quả định vị bằng VIO với độ trôi nhỏ, và khi có GPS trở lại, hệ tự động hiệu chỉnh để loại đi sai số tích lũy, khôi phục độ chính xác vốn có.

5.4.2. Thử nghiệm trên đường phố

Nhằm thúc đẩy giới hạn của hệ thống, học viên thực hiện thử nghiệm đi trên đường phố với môi trường đặc trưng động cao. Nhiều loại phương tiện liên tục chạy qua tầm nhìn của camera như trong Hình 5.41, Hình 5.42 và Hình 5.43. Quãng đường thử nghiệm dài khoảng 3km, tạo thành một vòng kín qua 4 đoạn đường xung quanh Đại học Bách Khoa TP.HCM. Thời gian di chuyển trong 50 phút với vận tốc trung bình 1m/s. Do môi trường có nhiều vật che chắn như nhà cửa, cây cối nên việc triển khai hệ GPS-RTK để đo vị trí chuẩn là không khả thi. Thay vào đó, vị trí chuẩn được

lấy từ hệ SPAN. Nhờ việc tích hợp chặt GPS/INS/Wheel, hệ SPAN có thể hoạt động trong môi trường bị che chắn một phần với sai số trung bình là 1.2m.



Hình 5.41. Đoạn đường Lý Thường Kiệt trong thử nghiệm



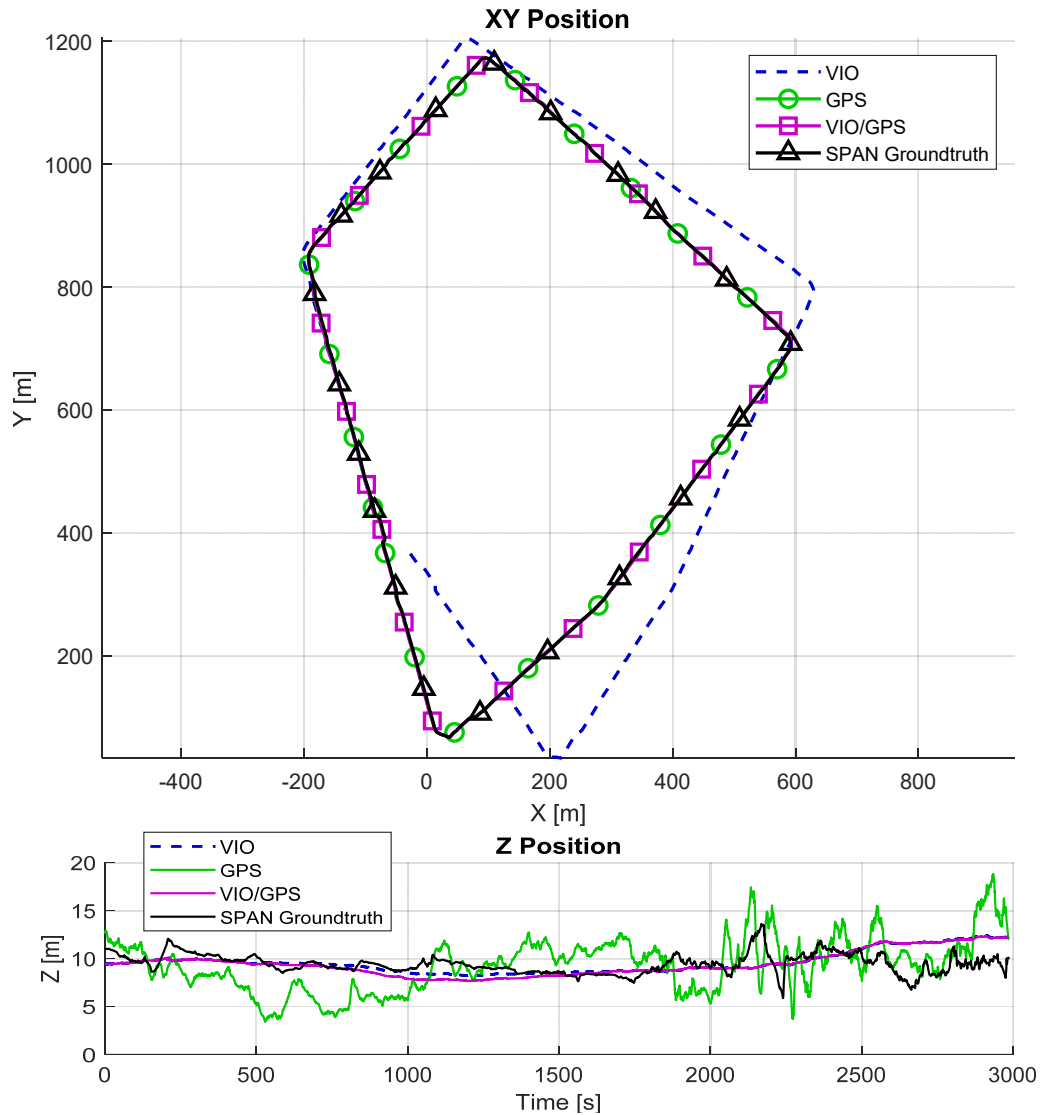
Hình 5.42. Đoạn đường Thành Thái trong thử nghiệm



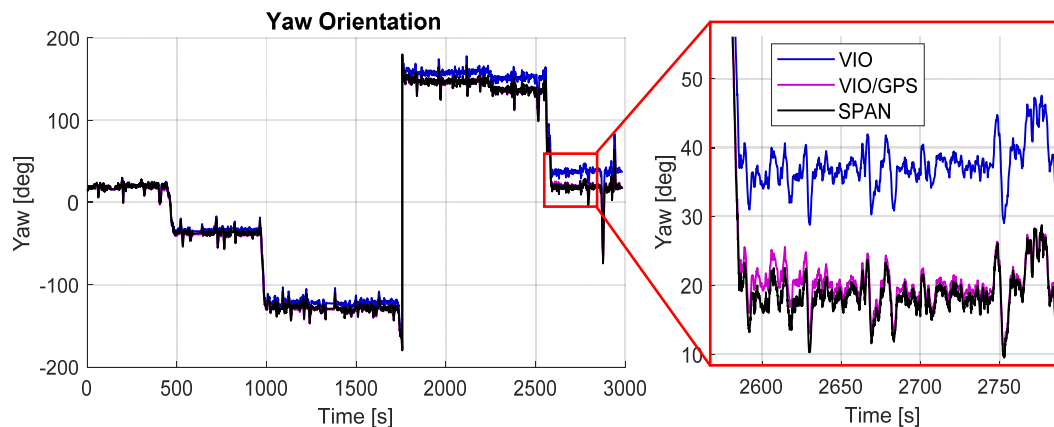
Hình 5.43. Đoạn đường Tô Hiến Thành trong thử nghiệm

Kết quả định vị trong Hình 5.44 cho thấy, quỹ đạo VIO sai lệch khá nhiều so với quỹ đạo chuẩn trên mặt phẳng XY, trong khi quỹ đạo GPS và VIO/GPS bám sát quỹ đạo chuẩn. Tuy nhiên, vị trí trục Z của VIO lại gần giá trị chuẩn hơn so với GPS, góp

phần vào độ chính xác của hệ tích hợp. Kết quả góc yaw của VIO/GPS cũng bám sát giá trị chuẩn và không trôi theo thời gian như VIO, như Hình 5.45.

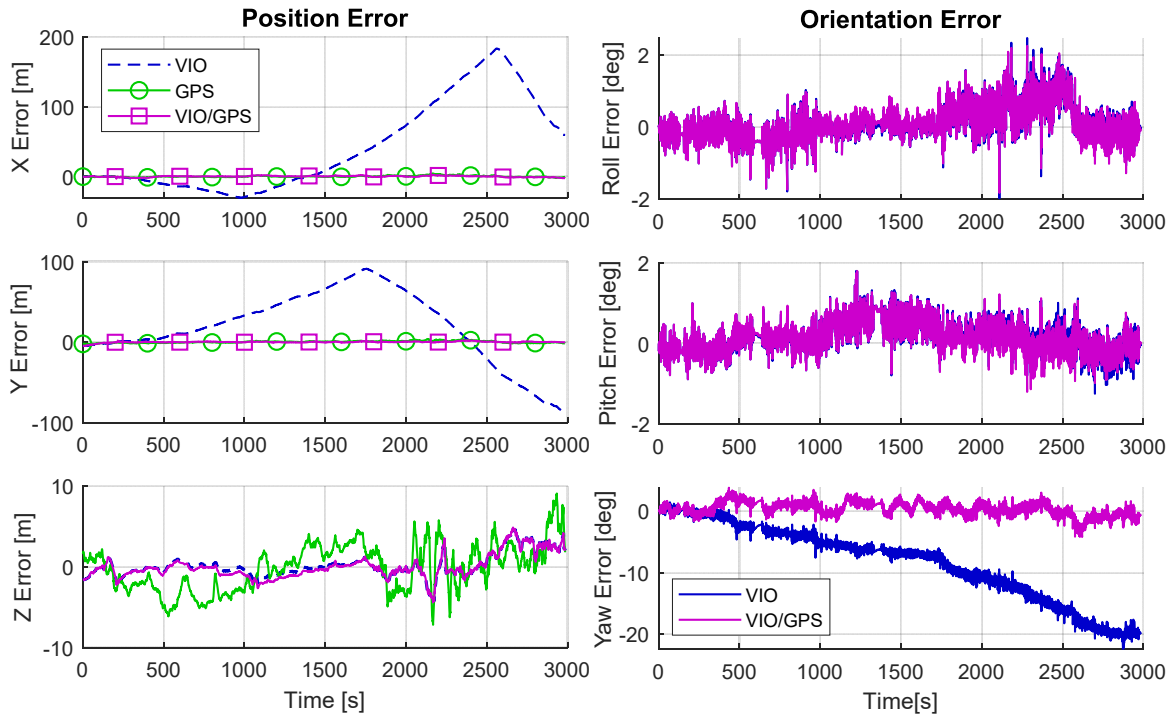


Hình 5.44. Kết quả định vị vị trí, tập dữ liệu Outdoor-Street

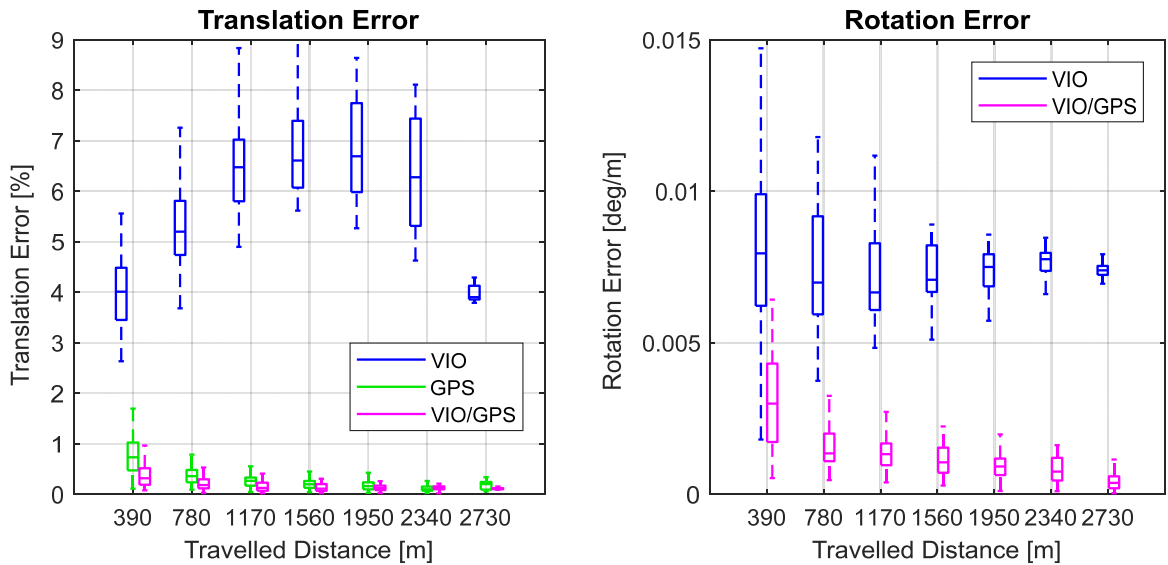


Hình 5.45. Kết quả định vị góc yaw, tập dữ liệu Outdoor-Street

Hình 5.46 cho thấy sai số VIO trên trục X,Y tăng khá lớn. Nguyên nhân là do diện tích vùng chuyển động thường xuyên chiếm đến 50% vùng ảnh camera, khiến thuật toán có ít điểm đặc trưng tĩnh để ước lượng. Mặt khác, nhiều điểm đặc trưng ở khoảng cách xa mà baseline của stereo camera lại khá ngắn (12cm) dẫn đến vị trí điểm trên hai ảnh stereo ít khác biệt, làm cho việc tái tạo landmark 3D có sai số lớn.



Hình 5.46. Sai số vị trí và góc xoay theo thời gian, tập dữ liệu Outdoor-Street



Hình 5.47. Sai số tương đối vị trí và góc xoay theo quãng đường, tập dữ liệu Outdoor-Street

Theo Hình 5.47, sai số vị trí tương đối của VIO dao động từ 4% đến 8% trên cả quãng đường, trong khi sai số của hệ tích hợp nhỏ hơn đáng kể và giảm dần đến mức 0.2% ở cuối quãng đường. Mặt khác, độ trôi góc của VIO là tương đối nhỏ, trên dưới $0.007^\circ/\text{m}$ do có ít chuyển động xoay trên quỹ đạo. Trong khi đó, độ trôi góc của hệ tích hợp giảm dần về 0 nhờ việc cập nhật GPS.

Tập dữ liệu	Hệ định vị	Sai số RMS vị trí [m]				
		X	Y	Z	2D	3D
Outdoor-Street	VIO	76.7479	48.2877	1.4273	90.6749	90.6862
	GPS	1.0626	1.2813	2.9017	1.6646	3.3453
	VIO/GPS	0.9331	0.8301	1.4827	1.2489	1.9386

Bảng 5.7. Sai số RMS vị trí, tập dữ liệu Outdoor-Street

Tập dữ liệu	Hệ định vị	Sai số RMS góc xoay [deg]			
		φ	θ	ψ	3D
Outdoor-Street	VIO	0.4267	0.3741	10.3811	10.3969
	VIO/GPS	0.4282	0.3618	1.1003	1.2349

Bảng 5.8. Sai số RMS góc xoay, tập dữ liệu Outdoor-Street

Theo Bảng 5.7, sai số vị trí tuyệt đối của hệ tích hợp VIO/GPS trên mặt phẳng 2D là 1.25m. Do bản thân dữ liệu trục Z của hệ SPAN cũng có dao động khá lớn trong những vùng GPS suy giảm nên giá trị sai số vị trí 3D chỉ mang tính tham khảo. Sai số tích hợp vị trí lớn hơn các thử nghiệm trước một phần vì chất lượng định vị VIO suy giảm, một phần vì vị trí chuẩn từ SPAN có độ chính xác kém hơn GPS-RTK. Theo Bảng 5.8, sai số góc yaw trong của hệ tích hợp đạt 1.1° , gần bằng thử nghiệm trong môi trường tĩnh, cho thấy độ chính xác góc yaw của hệ là ổn định.

5.4.3. Những trường hợp thị giác suy giảm và hướng khắc phục

Qua thử nghiệm, học viên nhận thấy hệ VIO bị suy giảm chất lượng trong một số điều kiện môi trường cực đoan, làm tăng sai số chung của hệ tích hợp VIO/GPS.

Về điều kiện sáng: Khi môi trường có cường độ ánh sáng mạnh vượt quá khả năng điều chỉnh màn trập của camera, ảnh sẽ xuất hiện những vùng trắng hoàn toàn như Hình 5.48. Hai trường hợp điển hình là mặt trời chiếu trực tiếp vào ảnh (a), và

mặt đường phản chiếu ánh nắng vào buổi trưa (b). Các đặc trưng bị mất đi trong vùng trắng làm cho thuật toán VIO thiếu dữ liệu để ước lượng chính xác, cũng như có nguy cơ phân kì nếu việc hiện tượng này kéo dài liên tục. Giải pháp tốt nhất là đưa dữ liệu GPS vào trong quá trình ước lượng VIO để tạo thành hệ tích hợp chặt, từ đó giữ ổn định các trạng thái ước lượng trong thời gian thị giác suy giảm. Một giải pháp khác là can thiệp điều chỉnh thời gian màn trập của camera theo hướng tối ưu hóa khả năng phát hiện điểm đặc trưng trên ảnh, như nghiên cứu tại [35].



(a)

(b)

Hình 5.48. Trường hợp ảnh bị chói sáng do mặt trời chiếu trực tiếp hoặc gián tiếp

Về chuyển động của môi trường: Thuật toán VIO dựa trên giả định môi trường là tĩnh để ước lượng chuyển động của robot. Thông thường, các vật chuyển động trong môi trường được loại đi nhờ quy trình RANSAC, trong đó đi tìm tập điểm đặc trưng có thay đổi chủ yếu trong ảnh và loại đi các điểm ngoại lệ. Tuy nhiên, cách làm này có nhược điểm là nếu đa số vật thể trong môi trường là động và có vận tốc đều nhau, thì RANSAC sẽ chọn chúng và loại đi những điểm tĩnh thiểu số. Khi đó, vị trí robot sẽ bị tính theo các vật thể động và sai lệch so với vị trí thực. Ví dụ trong Hình 5.49a, xe buýt chuyển động từ trái sang phải và choáng hết tầm nhìn camera. Thuật toán VIO căn cứ theo chiếc xe sẽ nhầm tưởng robot đang di chuyển từ phải sang trái. Còn trong trường hợp Hình 5.49b, dòng xe máy đông đúc chạy tốc độ đều nhau từ sau lên trước sẽ làm thuật toán nhầm tưởng robot đang đi lùi về sau. Giải pháp đề xuất là xây dựng một mạng CNN (Convolution Neural Network) thực hiện phân loại

và theo dõi vận tốc các phương tiện, từ đó loại trừ chuyển động trong thuật toán VIO cũng như phục vụ cho việc điều khiển về sau [36].



(a)



(b)

Hình 5.49. Trường hợp đa số vật thể trong tầm nhìn camera chuyển động đều

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Luận văn đã hoàn thành các mục tiêu đề ra. Luận văn đã trình bày thuật toán định vị tích hợp chặt thị giác lập thể và quán tính theo phương pháp tối ưu hóa phi tuyến, trong đó thực hiện đồng bộ chặt thời gian giữa stereo camera và IMU. Luận văn cũng xây dựng thuật toán định vị tích hợp lỏng hệ VIO và GPS sử dụng bộ lọc Kalman mở rộng, trong đó căn chỉnh tọa độ hai hệ thống theo phương pháp một góc xoay yaw. Phần cứng và phần mềm hệ thống đã được hoàn thiện. Nhiều thử nghiệm đã được thực hiện trong các điều kiện môi trường khác nhau, cho thấy các hệ thống hoạt động ổn định và chính xác. Các ưu điểm của hệ thống tích hợp VIO/GPS so với riêng hệ thống VIO cũng được chỉ rõ qua thực nghiệm.

Chất lượng hệ thống đạt yêu cầu trong môi trường tĩnh ở điều kiện trong nhà lẫn ngoài trời. Sai số tương đối hệ VIO đạt dưới 1% và $0.05^\circ/\text{m}$. Sai số tuyệt đối hệ tích hợp VIO/GPS đạt khoảng 0.5m và 1° . Trong môi trường động trên đường phố, hệ VIO có sai số tích lũy tăng lên đáng kể do thị giác suy giảm, tuy vậy hệ tích hợp VIO/GPS vẫn giữ được sai số ở mức thấp nhờ khả năng hiệu chỉnh độ trôi. Tần số cập nhật hệ thống lên đến 500Hz.

6.2. Hướng phát triển

Hệ thống có thể đạt độ chính xác cao hơn với thuật toán tích hợp chặt giữa VIO và GPS. Chuyển động trong môi trường có thể được loại bỏ bằng cách phân loại vật thể sử dụng mạng CNN. Mặt khác, hệ cũng có thể tích hợp thêm nhiều loại cảm biến như Lidar, Radar, siêu âm, Ultra-wideband,... để nâng cao chất lượng cũng như độ bền vững với điều kiện môi trường. Camera có thể thay bằng camera nhiệt để cho phép hệ VIO hoạt động trong bóng tối. Hệ thống định vị trong đề tài có thể dùng làm nền tảng để phát triển một phương tiện tự hành hoàn chỉnh.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Anh:

- [1] "Tesla Autopilot," Tesla Inc., [Online]. Available: <https://www.tesla.com/autopilot>.
- [2] "High great reveals world's first visual inertial odometry positioning drone," 2018. [Online]. Available: <https://www.designboom.com/technology/>.
- [3] A. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, 2007.
- [4] M. Bloesch, M. Burri, S. Omari, M. Hutter and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053-1072, 2017.
- [5] M. Brossard, S. Bonnabel and A. Barrau, "Unscented Kalman Filter on Lie Groups for Visual Inertial Odometry," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [6] M. Li, B. Kim and A. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013.
- [7] A. Geiger, J. Ziegler and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011.
- [8] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, 2015.
- [9] V. Usenko, J. Engel, J. Stuckler and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016.

- [10] H. Strasdat, J. Montiel and A. Davision, "Real-time monocular SLAM: Why filter?," in *IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, 2010.
- [11] M. Faessler, F. Fontana, C. Forster and D. Scaramuzza, "Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor," in *IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, 2015.
- [12] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014.
- [13] S. Lynen, M. Achtelik, S. Weiss, M. Chli and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to MAV navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.
- [14] L. Meier, P. Tanskanen, F. Fraundorfer and M. Pollefeys, "PIXHAWK: A system for autonomous flight using onboard computer vision," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.
- [15] M. Li and A. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690-711, 2013.
- [16] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, Australia, 2018.
- [17] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, 1988.
- [18] S. Leutenegger, M. Chli and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision*, Barcelona, Spain, 2011.
- [19] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381-395, 1981.

- [20] K. Madsen, H. Neilsen and O. Tingleff, "Methods for Non-linear least squares problems," Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004.
- [21] T. Barfoot, J. R. Forbes and P. T. Furgale, "Pose estimation using linearized rotations and quaternion algebra," *Acta Astronautica*, vol. 68, no. 1-2, pp. 101-112, 2011.
- [22] O. S.-H. a. M. Rabinovich, "Least-Squares Rigid Motion Using SVD," January 2017. [Online]. Available: <https://igl.ethz.ch/>. [Accessed May 2020].
- [23] A. P. A. Mohiner. G. Grewal, Kalman Filtering Theory and Practice Using MATLAB, John Wiley & Sons Inc., 2008.
- [24] J. Solà, "Quaternion kinematics for the error-state Kalman filter," October 2017. [Online]. Available: <http://www.iri.upc.edu/>. [Accessed May 2020].
- [25] "Bumblebee2 FireWire," Point Grey, [Online]. Available: www.flir.com.
- [26] "ADIS 16488A Datasheet," Analog Device, 2018. [Online]. Available: <https://www.analog.com/en/products/adis16488a.html>.
- [27] "NEO-M8N Datasheet," Ublox, 2014. [Online]. Available: <https://www.u-blox.com>.
- [28] "NANO HM-650," IEI, 2013. [Online]. Available: <https://ieiworld.com>.
- [29] "S82T Datasheet," South, [Online]. Available: southsurvey.co.za.
- [30] "SPAN-CPT Datasheet," Novatel, 2016. [Online]. Available: <https://novatel.com/>.
- [31] "Robot Operating System," [Online]. Available: <https://www.ros.org/>.
- [32] A. Geiger, P. Lenz and R. Urtansun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.
- [33] T. Qin, J. Pan, S. Cao and S. Shen, "A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors," 2019. [Online]. Available: <https://arxiv.org/abs/1901.03638>.

- [34] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004-1020, 2018.
- [35] Z. Zhang, C. Forster and D. Scaramuzza, "Active exposure control for robust visual odometry in HDR environments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.
- [36] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016.

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ CỦA TÁC GIẢ

1. Nguyen H. K. Tran and Vinh-Hao Nguyen, “A Tightly-Coupled Visual-Inertial System with Synchronized Time for Indoor Localization”, *International Journal of Mechanical Engineering and Robotics Research*. ISSN: 2278-0149. (Accepted).