

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

CAO HUỲNH PHÚC

**XÂY DỰNG THUẬT TOÁN ĐỊNH VỊ TÍCH HỢP IMU/LIDAR
DÙNG BỘ LỌC KALMAN LẶP**
**IMU/LIDAR INTEGRATED NAVIGATION ALGORITHM
USING ITERATED KALMAN FILTER**

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

LUẬN VĂN THẠC SĨ

TP. HỒ CHÍ MINH, tháng 12 năm 2023

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

CAO HUỲNH PHÚC

**XÂY DỰNG THUẬT TOÁN ĐỊNH VỊ TÍCH HỢP
IMU/LIDAR DÙNG BỘ LỌC KALMAN LẶP**
**IMU/LIDAR INTEGRATED NAVIGATION ALGORITHM
USING ITERATED KALMAN FILTER**

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

LUẬN VĂN THẠC SĨ

TP. HỒ CHÍ MINH, tháng 12 năm 2023

Công trình hoàn thành tại: Trường Đại học Bách Khoa - ĐHQG - HCM

Cán bộ hướng dẫn khoa học: TS. NGUYỄN VĨNH HÀO

Cán bộ chấm nhận xét 1 :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Cán bộ chấm nhận xét 2 :

(Ghi rõ họ, tên, học hàm, học vị và chữ ký)

Luận văn thạc sĩ được bảo vệ tại Trường Đại học Bách Khoa, ĐHQG Tp. HCM
ngày tháng năm

Thành phần Hội đồng đánh giá luận văn thạc sĩ gồm:

(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ luận văn thạc sĩ)

1.
2.
3.
4.
5.

Xác nhận của Chủ tịch Hội đồng đánh giá LV và Trưởng Khoa quản lý chuyên
ngành sau khi luận văn đã được sửa chữa (nếu có).

CHỦ TỊCH HỘI ĐỒNG

TRƯỞNG KHOA ĐIỆN – ĐIỆN TỬ

NHIỆM VỤ LUẬN VĂN THẠC SĨ

Họ tên học viên: CAO HUỲNH PHÚC

MSHV: 2070627

Ngày, tháng, năm sinh: 11/09/1998

Nơi sinh: TP. Hồ Chí Minh

Chuyên ngành: Kỹ thuật Điều khiển và Tự động hóa

Mã số: 8520216

I. TÊN ĐỀ TÀI:

XÂY DỰNG THUẬT TOÁN ĐỊNH VỊ TÍCH HỢP IMU/LIDAR DÙNG BỘ LỌC KALMAN LẮP

II. NHIỆM VỤ VÀ NỘI DUNG:

- Xây dựng thuật toán định vị tích hợp IMU và Lidar.
- Xây dựng phần cứng, phần mềm cho hệ thống thực nghiệm.
- Thực nghiệm và đánh giá hệ thống định vị.

III. NGÀY GIAO NHIỆM VỤ: 06/02/2023

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 12/2023

V. CÁN BỘ HƯỚNG DẪN: TS. NGUYỄN VĨNH HẢO

CÁN BỘ HƯỚNG DẪN

Tp. HCM, ngày tháng năm 20....

CHỦ NHIỆM BỘ MÔN ĐÀO TẠO

TRƯỜNG KHOA ĐIỆN – ĐIỆN TỬ

LỜI CẢM ƠN

Đầu tiên, em xin gửi lời cảm ơn đến các quý thầy cô trong khoa Điện – Điện Tử và bộ môn Điều khiển Tự động trường Đại học Bách Khoa đã trang bị những kiến thức bổ ích cùng những kinh nghiệm quý báu trong suốt những năm đại học và cao học.

Em xin gửi lời cảm ơn đến thầy Nguyễn Vĩnh Hảo đã định hướng, hướng dẫn và hỗ trợ em trong suốt hành trình thực hiện luận văn.

Cảm ơn bạn Đinh Quang Vũ đã đồng hành trong suốt chặng đường đại học và cao học. Cảm ơn các em Nguyễn Thanh Toàn, Nguyễn Minh Quang, Trương Minh Mẫn, Thi Công Thành, Phí Thành Danh và Nguyễn Ngọc Thanh Tuấn đã hỗ trợ hết mình. Em xin cảm ơn hai anh Trần Hoàng Khôi Nguyên và Trần Quốc Tiến Dũng đã hỗ trợ em về sử dụng thiết bị, cơ sở vật chất phòng thí nghiệm.

Mình cũng xin gửi lời cảm ơn đến bạn bè, các anh em thân thiết, các anh chị bạn bè đồng nghiệp đã luôn sát cánh, giúp đỡ tận tình trong suốt hành trình tại Đại học Bách Khoa.

Lời cảm ơn cuối cùng con xin gửi lời cảm ơn đến gia đình thân yêu, đến bố mẹ và em trai đã luôn là những người lặng lẽ động viên, là hậu phương vững chắc, là động lực để con nỗ lực hết sức mình.

TÓM TẮT LUẬN VĂN THẠC SĨ

Định vị là một module quan trọng không thể thiếu trong các ứng dụng liên quan đến robot tự hành hoặc xe tự hành. Bên cạnh GPS, các hệ định vị quán tính cũng đang dần được sử dụng rộng rãi vì có thể cung cấp thông tin định vị cục bộ chính xác trong khoảng thời gian ngắn khi mà bộ thu GPS không thể nhận được tín hiệu do bị che chắn. Hệ định vị quán tính thường bao gồm IMU và các cảm biến hỗ trợ đi kèm như Lidar, Camera, Encoder, Radar sóng âm hoặc cảm biến đo chiều cao Altimeter. Luận văn thực hiện hai nhiệm vụ chính là xây dựng thuật toán định vị kết hợp IMU và Lidar sử dụng bộ lọc Kalman lặp IEKF, và triển khai hệ thống thực nghiệm để đánh giá thuật toán. Nhờ giải thuật Kalman lặp và dữ liệu trả về từ Lidar, sai số tích lũy của hệ định vị quán tính IMU sẽ được cải thiện. Các thí nghiệm với các môi trường và quỹ đạo khác nhau được thực hiện để chứng minh hiệu quả của bộ định vị tích hợp IMU/Lidar. Bên cạnh đó, một thuật toán tích hợp IMU/Lidar/Encoder cũng sẽ được phát triển dựa trên thuật toán tích hợp Lidar/IMU ban đầu nhằm mục đích cải thiện chất lượng bộ định vị.

ABSTRACT

Navigation is an important and indispensable module in applications related to autonomous robots or autonomous vehicles. Besides GPS, inertial navigation systems are also gradually being widely used because they can provide reliable local positioning information in a short period of time when the GPS receiver cannot receive the signal due to masking. Inertial navigation system usually includes IMU and supporting sensors such as Lidar, Camera, Encoder, Radar or Altimeter. The thesis focuses two main tasks: building a positioning algorithm combining IMU and Lidar using IEKF iterated Kalman filter, and implementing the experiment system to evaluate the algorithm. Thanks to the iterated Kalman algorithm and the data returned from the Lidar, the cumulative error of the inertial navigation system using IMU will be improved. Experiments with different environments and trajectories are performed to demonstrate the effectiveness of the IMU/Lidar integrated system. In addition, IMU/Lidar/Encoder integrated system is developed based on the orginal algorithm with the aim of improving the quality of the navigation system.

LỜI CAM KẾT

Tôi xin cam kết luận văn này là do tôi thực hiện. Nội dung tham khảo từ các tác giả đều được trích dẫn đầy đủ. Tất cả kết quả thực nghiệm đều là thật và do chính tôi làm ra, không sao chép từ bất cứ công trình nào khác. Nếu có điều nào không đúng, tôi xin chịu trách nhiệm trước bộ môn và nhà trường.

CAO HUỲNH PHÚC

MỤC LỤC

DANH MỤC HÌNH VẼ.....	4
DANH MỤC BẢNG.....	9
DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT	10
Chương 1. GIỚI THIỆU	12
1.1. Mục đích nghiên cứu.....	12
1.2. Tổng quan về tình hình nghiên cứu	13
1.3. Nội dung và mục tiêu luận văn	17
1.3.1. Nội dung luận văn.....	17
1.3.2. Mục tiêu luận văn	17
1.4. Cấu trúc luận văn	17
Chương 2. CƠ SỞ LÝ THUYẾT	18
2.1. Nguyên lý hoạt động của các loại cảm biến	18
2.1.1. Cảm biến IMU	18
2.1.1.1. Cảm biến gia tốc.....	19
2.1.1.2. Cảm biến vận tốc góc	21
2.1.2. Cảm biến Lidar	24
2.1.3. Cảm biến Encoder.....	29
2.2. Các hệ tọa độ và các biến trạng thái của hệ thống	31
2.2.1. Các hệ tọa độ	31
2.2.2. Các biến trạng thái	33
2.3. Lý thuyết Lie	34
2.3.1. Tổng quan	34
2.3.2. Phép \oplus và \ominus trong $SO(3)$	35
2.3.3. Đạo hàm trong $SO(3)$	36
2.3.4. Đa tạp phức hợp các nhóm Lie	37
2.3.5. Phép \oplus và \ominus trong không gian trạng thái	37
2.4. Bộ lọc Kalman lặp IEKF (Iterated Extended Kalman Filter) [44]	38
2.4.1. Giới thiệu	38

2.4.2. Mô hình toán hệ thống	38
2.4.3. Dự đoán trạng thái	39
2.4.4. Cập nhật trạng thái	39
Chương 3. THUẬT TOÁN ĐỊNH VỊ TÍCH HỢP IMU/LIDAR	42
3.1. Tổng quan về thuật toán	42
3.2. Định vị quán tính sử dụng IMU	43
3.2.1. Mô hình phi tuyến IMU	43
3.2.2. Tính toán trạng thái	44
3.2.3. Tính toán covariance (ma trận hiệp phương sai)	45
3.3. Cập nhật trạng thái từ dữ liệu của Lidar	46
3.3.1. Phương pháp đồng bộ dữ liệu cảm biến Lidar	46
3.3.2. Tính toán thặng dư	48
3.3.3. Mô hình đo lường	49
3.4. Cập nhật bản đồ	51
3.5. Thuật toán kết hợp Lidar/IMU/Encoder	53
Chương 4. HỆ THỐNG PHẦN CỨNG VÀ PHẦN MỀM	56
4.1. Cấu trúc phần cứng	56
4.2. Cấu trúc phần mềm	60
Chương 5. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ	63
5.1. Tổng quan	63
5.2. Phương pháp đánh giá sai số	65
5.3. Kết quả thực nghiệm indoor	66
5.3.1. Tập dữ liệu di chuyển nhanh	67
5.3.2. Tập dữ liệu di chuyển chậm	69
5.4. Kết quả thực nghiệm outdoor	71
5.4.1. Tập dữ liệu di chuyển theo hình tròn	72
5.4.1.1. Di chuyển nhanh	72
5.4.1.2. Di chuyển chậm	76
5.4.2. Tập dữ liệu di chuyển theo hình chữ nhật	79

5.4.2.1. Di chuyển nhanh.....	79
5.4.2.2. Di chuyển chậm.....	82
5.4.3. Tập dữ liệu di chuyển xung quanh khuôn viên trường HCMUT	85
5.5. Tập dữ liệu có encoder.....	90
5.6. Nhận xét	95
Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	96
6.1. Kết luận	96
6.2. Hướng phát triển	96
TÀI LIỆU THAM KHẢO.....	98

DANH MỤC HÌNH VẼ

Hình 1.1. UGV Husky A200 (a) thu thập dữ liệu Lidar, ước lượng vị trí dùng phương pháp feature-based scan matching cho ứng dụng outdoor (b) và indoor (c), sai số 0.44m trên tổng quãng đường 440m [2].....	14
Hình 1.2. Xe hơi trang bị cảm biến (a) thu thập dữ liệu Lidar, ước lượng vị trí dùng phương pháp tích hợp chặt (b), sai số 0.262m trên tổng quãng đường 1.21km [10].....	15
Hình 2.1. Một số sản phẩm IMU của hãng Analog Device và XSENS [18], [19]	18
Hình 2.2. Mô hình đo gia tốc đơn trực (a) và mô hình đo gia tốc ba trực (b) [20]	19
Hình 2.3. Mô hình đo gia tốc của cảm biến gia tốc dạng MEMS [21]	20
Hình 2.4. Minh họa về lực tác động lên cảm biến gia tốc khi nằm yên [23]	21
Hình 2.5. Minh họa về lực Coriolis [20].....	22
Hình 2.6. Lực tác dụng lên cảm biến khi có vận tốc góc (a) và khi chỉ có gia tốc (b) ..	23
Hình 2.7. Minh họa ba góc roll, pitch, yaw của vật thể [26]	24
Hình 2.8. Lidar được ứng dụng trong công nghệ ô tô [28], [29]	25
Hình 2.9. Minh họa hoạt động của cảm biến Lidar [30]	25
Hình 2.10. FOV của Lidar [32]	26
Hình 2.11. Cơ chế hoạt động của Lidar dạng cơ [34]	27
Hình 2.12. Các dòng sản phẩm Lidar cơ của Velodyne và hình dạng point cloud của nó [35], [36].....	27
Hình 2.13. Cơ chế hoạt động của Lidar dạng MEMS [37]	28
Hình 2.14. AVIA Lidar của hãng Livox và hình dạng point cloud của nó [38], [39] ..	28
Hình 2.15. Encoder của hãng Autonics (a) và hãng Omron (b) [40], [41]	29
Hình 2.16. Nguyên lý hoạt động của Encoder quang [42].....	30
Hình 2.17. Nguyên lý hoạt động của Encoder quang [43].....	30
Hình 2.18. Minh họa hệ tọa độ gắn liền với IMU và Lidar	32
Hình 2.19. Minh họa về hệ tọa độ Global và hệ tọa độ gắn liền với IMU.....	32

Hình 2.20. Minh họa đa tạp tron \mathcal{M} và không gian tiếp tuyến $T\mathcal{X}\mathcal{M}$ của \mathcal{M} tại \mathcal{X} [45]	34
Hình 3.1. Sơ đồ thuật toán định vị tích hợp IMU/Lidar.....	42
Hình 3.2. Thời gian lấy mẫu IMU và các điểm trong một frame truyền Lidar [7].....	48
Hình 3.3. Minh họa tính toán thặng dư từ điểm đặc trưng của Lidar [50].....	49
Hình 3.4. Minh họa quá trình cập nhật bản đồ.....	52
Hình 3.5. Sơ đồ thuật toán định vị tích hợp IMU/Lidar.....	53
Hình 3.6. Minh họa hệ IMU và Encoder.....	54
Hình 4.1. Sơ đồ kết nối phần cứng hệ thống thực nghiệm.....	56
Hình 4.2. Mô hình phần cứng hệ thống thực nghiệm	57
Hình 4.3. Cảm biến IMU ADIS16488 [56]	58
Hình 4.4. Cảm biến Lidar Velodyne VLP-16 [29]	58
Hình 4.5. Stereo camera Point Grey Bumblebee2 [59].....	59
Hình 4.6. Máy tính nhúng IEI NANO-HM650.....	59
Hình 4.7. GPS RTK South S82-T	60
Hình 4.8. Cơ chế hoạt động Public and Subscribe của ROS.....	61
Hình 4.9. Cấu trúc phần mềm	62
Hình 4.10. Giao diện Rviz.....	62
Hình 5.1. Bố trí hệ thống thực nghiệm.....	64
Hình 5.2. Bên trong tòa nhà A4	66
Hình 5.3. Bản đồ dựng được bên trong tòa nhà A4	66
Hình 5.4. Quỹ đạo 2D tập dữ liệu indoor di chuyển nhanh	67
Hình 5.5. Quỹ đạo IMU/Camera bị lệch do nhóm sinh viên di chuyển	68
Hình 5.6. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển nhanh	68
Hình 5.7. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển nhanh	69
Hình 5.8. Quỹ đạo 2D tập dữ liệu indoor di chuyển chậm	70

Hình 5.9. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển chậm	70
Hình 5.10. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển chậm	71
Hình 5.11. Phía trước tòa nhà A2.....	71
Hình 5.12. Bản đồ dựng được phía trước tòa nhà A2	72
Hình 5.13. Quỹ đạo 2D tập dữ liệu outdoor hình tròn di chuyển nhanh	73
Hình 5.14. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh.....	74
Hình 5.15. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh	74
Hình 5.16. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh	75
Hình 5.17. Phần trăm sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh	75
Hình 5.18. Quỹ đạo 2D tập dữ liệu outdoor hình tròn di chuyển chậm.....	76
Hình 5.19. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm	77
Hình 5.20. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm	77
Hình 5.21. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm	78
Hình 5.22. Phần trăm sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm	78
Hình 5.23. Quỹ đạo 2D tập dữ liệu outdoor hình chữ nhật di chuyển nhanh	79
Hình 5.24. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh	80

Hình 5.25. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh	80
Hình 5.26. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh.....	81
Hình 5.27. Phần trăm sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh.....	81
Hình 5.28. Quỹ đạo 2D tập dữ liệu outdoor hình chữ nhật di chuyển chậm	82
Hình 5.29. Dữ liệu vị trí trực z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm	83
Hình 5.30. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm	83
Hình 5.31. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm.....	84
Hình 5.32. Phần trăm sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm	84
Hình 5.33. Khuôn viên trường Đại học Bách Khoa TP.HCM	85
Hình 5.34. Bản đồ dựng được của khuôn viên trường HCMUT	85
Hình 5.35. Một phần bản đồ khu vực B9, B8, B11, C3, C4, C1 trường HCMUT	86
Hình 5.36. Một phần bản đồ khu vực B6, B4, B3, B2, A2, A5 trường HCMUT	86
Hình 5.37. Quỹ đạo 2D tập dữ liệu outdoor khuôn viên HCMUT	87
Hình 5.38. Dữ liệu vị trí trực z của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT	88
Hình 5.39. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT	88
Hình 5.40. Dữ liệu góc hướng của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT	89
Hình 5.41. Bố trí cảm biến trong tập dữ liệu Kaist [52]	90
Hình 5.42. Vị trí của khu vực khảo sát [52].....	91

Hình 5.43. Bản đồ dựng được tại khu vực khảo sát.....	91
Hình 5.44. Quỹ đạo 2D tập dữ liệu Kaist.....	92
Hình 5.45. Dữ liệu vị trí trực z của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu Kaist ...	93
Hình 5.46. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu Kaist	93
Hình 5.47. Dữ liệu góc hướng của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu outdoor Kaist	94
Hình 5.48. Phần trăm sai số theo quãng đường của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu Kaist	94

DANH MỤC BẢNG

Bảng 5.1. Bảng thông số sử dụng cho các tập dữ liệu tại HCMUT	63
Bảng 5.2. Thông tin tập dữ liệu indoor di chuyển nhanh.....	67
Bảng 5.3. Kết quả thực nghiệm tập dữ liệu indoor di chuyển nhanh.....	67
Bảng 5.4. Thông tin tập dữ liệu indoor di chuyển nhanh.....	69
Bảng 5.5. Kết quả thực nghiệm tập dữ liệu indoor di chuyển nhanh.....	69
Bảng 5.6. Thông tin tập dữ liệu outdoor, hình tròn, di chuyển nhanh	72
Bảng 5.7. Kết quả thực nghiệm tập dữ liệu outdoor, hình tròn, di chuyển chậm	72
Bảng 5.8. Thông tin tập dữ liệu outdoor, hình tròn, di chuyển chậm	76
Bảng 5.9. Kết quả thực nghiệm tập dữ liệu outdoor, hình tròn, di chuyển chậm	76
Bảng 5.10. Thông tin tập dữ liệu outdoor, hình chữ nhật, di chuyển nhanh.....	79
Bảng 5.11. Kết quả thực nghiệm tập dữ liệu outdoor, hình chữ nhật, di chuyển nhanh	79
Bảng 5.12. Thông tin tập dữ liệu outdoor, hình chữ nhật, di chuyển chậm.....	82
Bảng 5.13. Kết quả thực nghiệm tập dữ liệu outdoor, hình chữ nhật, di chuyển chậm.	82
Bảng 5.14. Thông tin tập dữ liệu khuôn viên trường HCMUT	87
Bảng 5.15. Kết quả thực nghiệm tập dữ liệu khuôn viên trường HCMUT	87
Bảng 5.16. Bảng thông số sử dụng cho các tập dữ liệu Kaist [52]	90
Bảng 5.17. Thông tin tập dữ liệu Kaist	92
Bảng 5.18. Kết quả thực nghiệm tập dữ liệu Kaist	92

DANH MỤC KÝ HIỆU VÀ TỪ VIẾT TẮT

Ký hiệu và chữ viết tắt	Giải thích ý nghĩa	Ghi chú
IEKF	Iterated Extended Kalman Filter	
EKF	Extended Kalman Filter	
IMU	Cảm biến đo lường quán tính (Inertial Measurement Unit)	
MEMS	Hệ thống vi cơ điện tử (MicroElectroMechanical System)	
DCM	Ma trận chuyển đổi cosine (Direct cosine matrix)	
Lidar	Light Detection And Ranging	
VĐK	Vị điều khiển	
UGV	Unmanned ground vehicle	
UAV	Unmanned aerial vehicle	
t_k	Thời điểm kết thúc của lần scan thứ k của Lidar	
τ_i	Thời điểm lấy mẫu thứ i của IMU trong một lần scan của Lidar	
ρ_j	Thời điểm lấy mẫu của feature point thứ j trong một lần scan của Lidar	
I_i, I_j, I_k	IMU body frame ở các thời điểm τ_i, ρ_j, t_k	
L_j, L_k	Lidar body frame ở các thời điểm ρ_j và t_k	
x, \hat{x}, \bar{x}	Giá trị đúng của x , giá trị của x trong quá trình định vị sử dụng IMU, giá trị của x sau cập nhật	
\tilde{x}	Sai số giữa giá trị đúng x và giá trị ước lượng \bar{x}	
\hat{x}^κ	Giá trị cập nhật thứ κ trong quá trình lặp của bộ lọc IEKF	
x_i, x_j, x_k	Vector x tại các thời điểm τ_i, ρ_j, t_k	
\tilde{x}_j	Chênh lệch của x_j so với x_k trong quá trình đồng bộ dữ liệu của Lidar	
$[\omega]_\wedge$	Ma trận phản đối xứng của vector $\omega = [\omega_x \quad \omega_y \quad \omega_z] \in \mathbb{R}^3$, $[\omega]_\wedge = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$	

a_m, ω_m	Giá trị đo gia tốc thẳng và vận tốc góc của IMU	
a_e	Gia tốc gây ra bởi chuyển động của vật thể	
$^G g$	Vector gia tốc trọng trường Trái Đất trong hệ tọa độ G	
$^G R_I$	Ma trận xoay (DCM) của hệ tọa độ I so với hệ tọa độ G	
$\ x\ $	Độ lớn vector x	

Chương 1. GIỚI THIỆU

1.1. Mục đích nghiên cứu

Định vị là một phần không thể thiếu khi muốn điều khiển hoặc giám sát bất kỳ đối tượng nào đặc biệt là trong những môi trường bất định, nguy hiểm, những nơi không thể sử dụng các hệ thống định vị thông thường như GPS. Trong những trường hợp như vậy, một bộ định vị sử dụng quán tính là một sự lựa chọn hợp lý vì các bộ định vị này thường độc lập với các thiết bị khác, dựa vào chuyển động của đối tượng, robot (vận tốc góc, gia tốc) để ước lượng trạng thái, vị trí, góc hướng. Bên cạnh đó IMU hay cảm biến quán tính thường được sử dụng trong các bộ định vị này có kích thước nhỏ gọn, giá thành rẻ, dễ dàng tích hợp lên các robot như phương tiện tự hành, UAV (Unmanned aerial vehicle),...

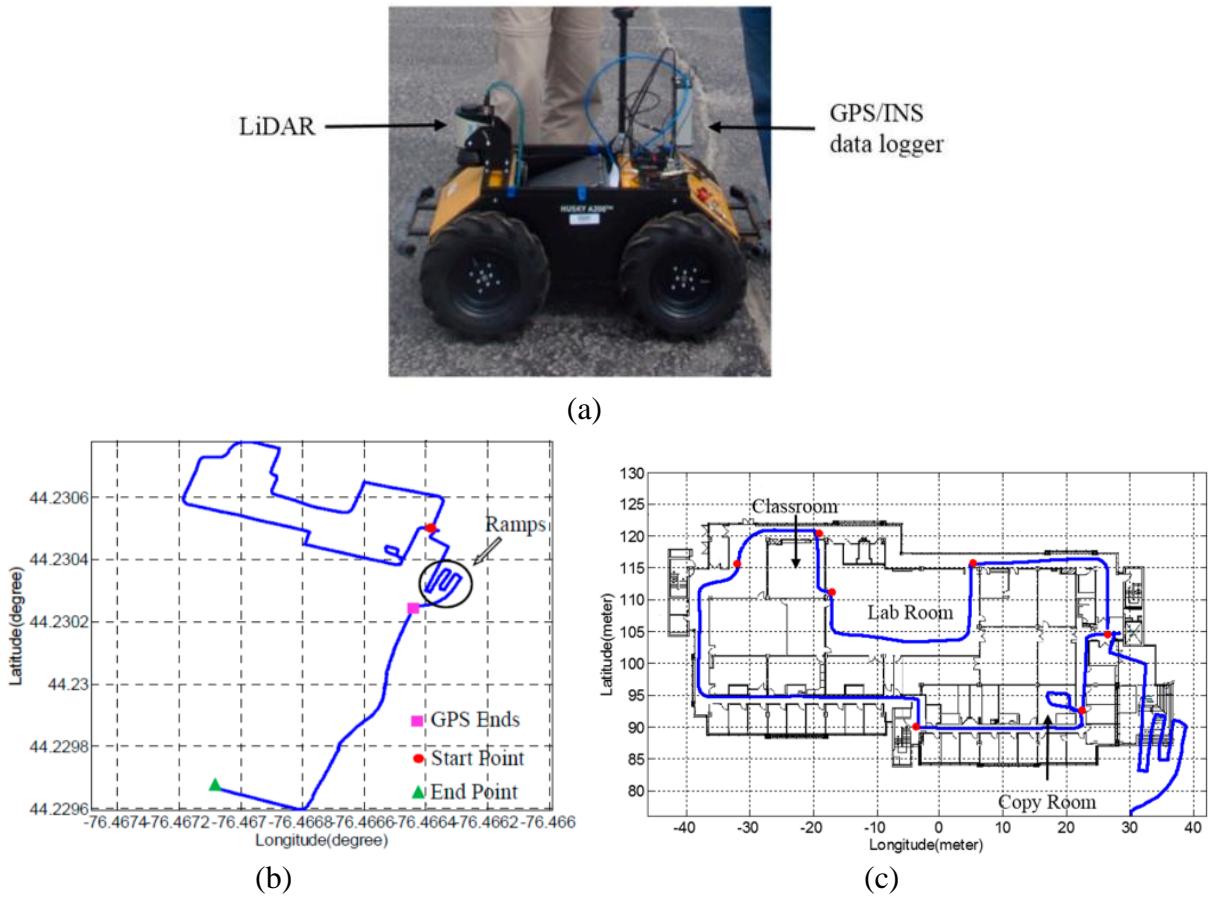
Tuy IMU có thể ước lượng được các vị trí và góc hướng từ gia tốc và vận tốc góc nhưng các cảm biến IMU dễ bị ảnh hưởng bởi nhiều nén sai số ước lượng sẽ bị tích lũy theo thời gian, độ chính xác chỉ có thể duy trì trong thời gian ngắn. Chính vì vậy định vị quán tính thường được cải thiện độ chính xác bằng sự hỗ trợ của các cảm biến khác. Một trong số đó là Lidar, một loại cảm biến đo khoảng cách tới mục tiêu bằng cách chiếu sáng mục tiêu đó bằng một tia laser xung quanh và đo thời gian nhận xung phản xạ. Lidar hiện nay là một cảm biến được sử dụng rộng rãi trong các phương tiện tự hành với nhiệm vụ hỗ trợ định vị và xây dựng bản đồ. Lidar tuy có độ chính xác ổn định theo thời gian, ngoài ước lượng vị trí còn có thể dùng để dựng bản đồ 3D nhưng thời gian để quét hết các điểm trong không gian thường rất lớn. Lúc này IMU sẽ phát huy vai trò của mình là duy trì được độ chính xác trong thời gian ngắn để Lidar kịp trả về tất cả các dữ liệu.

Trong luận văn này, hệ thống định vị tích hợp IMU và Lidar sẽ được xây dựng dựa trên bộ lọc Kalman lặp IEKF (Iterated Extended Kalman Filter). Các biến trạng thái gồm có vị trí, vận tốc, góc hướng và độ lệch của các cảm biến. Hệ thống định vị tích hợp này cung cấp dữ liệu về vị trí có thể được sử dụng cho các robot hoặc phương tiện tự hành

trong các môi trường trong nhà hoặc ngoài trời. Để cải thiện độ chính xác vị trí, thuật toán định vị tích hợp IMU/Lidar/Encoder cũng sẽ được phát triển.

1.2. Tổng quan về tình hình nghiên cứu

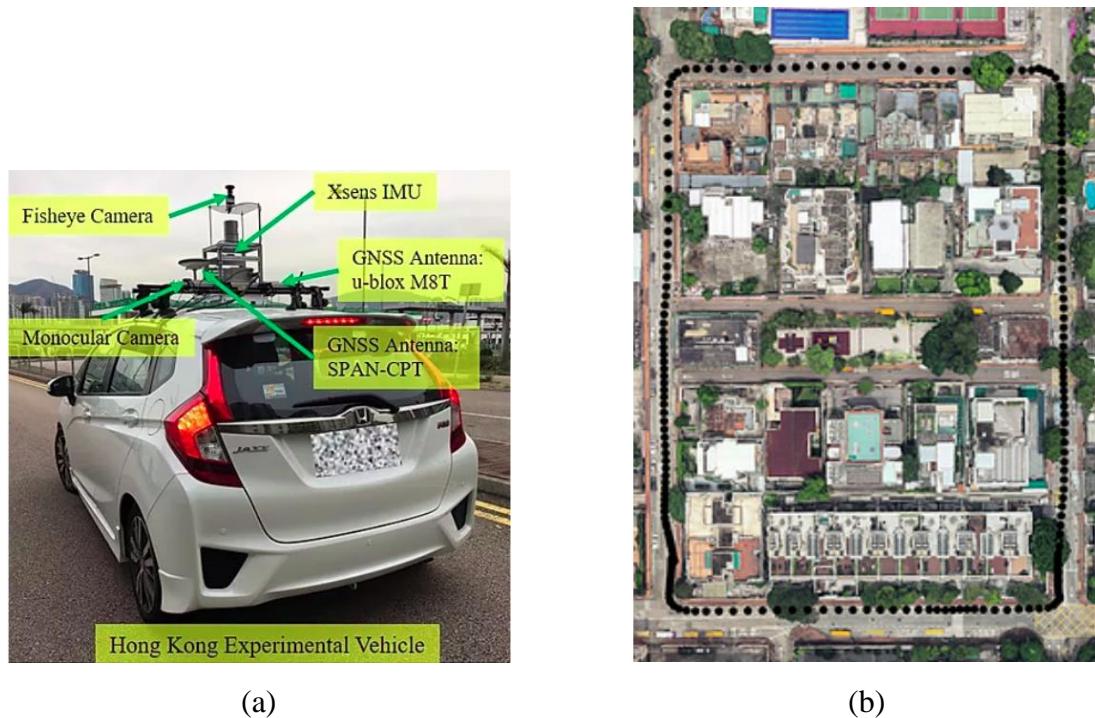
IMU và Lidar được sử dụng kết hợp với nhau để giảm sai số tích lũy khi sử dụng bộ định vị quán tính. Ngoài ra các phương pháp sử dụng Lidar sẽ cho ra kết quả kém trong môi trường ít đặc trưng, không đủ phong phú về mặt hình học, việc kết hợp với IMU giúp cải thiện chất lượng ước lượng [1]. Xét về mức độ tích hợp, có hai cách tiếp cận là tích hợp lỏng và tích hợp chặt. Đối với tích hợp lỏng, dữ liệu của IMU và Lidar sẽ được xử lý độc lập và thường kết hợp với nhau ở gian đoạn cuối để hiệu chỉnh sai số. Để xử lý dữ liệu thô từ Lidar, một phương pháp thường dùng là feature-based scan matching để tính độ lệch vị trí, góc hướng của đối tượng giữa hai chùm tia quét dựa vào các đường thẳng trùng khớp nhau giữa hai chùm tia quét đó, sau đó độ chênh này có thể được đưa vào bộ lọc Kalman để cập nhật các sai số trạng thái cho bộ định vị quán tính [2]. Trong khi đó Zhang và các cộng sự [3] sử dụng IMU để tiền xử lý data từ Lidar, hiệu chỉnh các dữ liệu về hệ tọa độ gắn với Lidar tại thời điểm bắt đầu một chu kỳ quét, sau đó họ sử dụng phương pháp ICP [4] để xác định chênh lệch chuyển động giữa hai Lidar scan frame sao cho cực tiểu khoảng cách giữa điểm đặc trưng trong scan frame hiện tại và đường thẳng hoặc mặt phẳng tương ứng trong scan frame trước đó, các nhóm tác giả [5] và [6] cũng có cách tiếp cận tương tự. Phương pháp tích hợp lỏng thường có khối lượng tính toán ít, quá trình xử lý dữ liệu Lidar độc lập với quá trình kết hợp với IMU. Phương pháp này nhìn chung khá phụ thuộc vào vị trí và góc hướng được tính toán từ Lidar, không xét đến sự liên quan của các biến trạng thái khác (ví dụ như vận tốc) [7]. Vì vậy trong một số trường hợp khi các đặc trưng hình học của môi trường (cạnh đường thẳng và mặt phẳng) không đủ dẫn đến các phép đo Lidar bị “suy thoái” và không thể tin tưởng, độ chính xác sẽ không thể duy trì.



Hình 1.1. UGV Husky A200 (a) thu thập dữ liệu Lidar, ước lượng vị trí dùng phương pháp feature-based scan matching cho ứng dụng outdoor (b) và indoor (c), sai số 0.44m trên tổng quãng đường 440m [2]

Phương pháp tiếp cận thứ hai là phương pháp tích hợp chặt. Phương pháp này sử dụng trực tiếp dữ liệu thô của Lidar kết hợp trực tiếp với IMU, chia làm hai hướng tiếp cận là dựa trên tối ưu hóa và dựa trên bộ lọc. Đối với phương pháp tối ưu hóa, dữ liệu từ IMU được tích phân trong khoảng thời gian giữa hai chùm tia quét liên tiếp để tính toán chuyển động tương đối giữa hai Lidar frame, bên cạnh đó dữ liệu thô Lidar cũng sẽ được xử lý với các phương pháp scan matching đã được đề cập trước đó. Các ràng buộc về chuyển động có được từ IMU và Lidar sẽ được tối ưu hóa để ra được kết quả cuối cùng. Nhóm tác giả [8] bổ sung thêm ràng buộc có từ GPS để tăng độ tin cậy cho kết quả ước lượng, ngoài ra chỉ sử dụng các keyframe có được từ chênh lệch chuyển động tối thiểu

để giảm khối lượng tính toán. LILI-OM [9] cũng sử dụng keyframe nhưng keyframe được lựa chọn dựa vào tỉ lệ trùng lắp data với bản đồ cục bộ và khoảng thời gian tối thiểu giữa hai chùm tia quét liên tiếp. Jiachen Zhang và Weisong Wen [10] bổ sung thêm ràng buộc liên quan giữa các trạng thái ước lượng và vị trí, vận tốc, góc hướng có được từ tích phân giá trị đo của IMU. Đối với phương pháp bộ lọc, các trạng thái được ước lượng dựa trên giá trị đo và sự bất định, không chắc chắn của giá trị đo đó, cụ thể các giá trị đo được lấy từ Lidar [11]. Joel A. Hesch [12] giới thiệu lidar odometry sử dụng bộ lọc Kalman mở rộng (EKF) nhưng phương pháp được đề xuất quá phụ thuộc vào đặc trưng của tường, trần nhà và nền nhà, chỉ thích hợp indoor. Các tác giả LINS [13] giới thiệu phương pháp ước lượng sử dụng bộ lọc Kalman lắp và cảm biến Lidar 3D, phù hợp cho các ứng dụng cả indoor và outdoor.



Hình 1.2. Xe hơi trang bị cảm biến (a) thu thập dữ liệu Lidar, ước lượng vị trí dùng phương pháp tích hợp chặt (b), sai số 0.262m trên tổng quãng đường 1.21km [10]

Các phương pháp tích hợp chặt cho ra kết quả ước lượng tốt hơn các phương pháp tích hợp lỏng [8] nhưng bù lại khối lượng tính toán sẽ tăng đáng kể đặc biệt là hướng sử dụng tối ưu hóa vì phương pháp này yêu cầu phải tối ưu hóa một lượng lớn các ràng buộc, bao gồm các ràng buộc trong quá khứ [13]. Chính vì vậy đề tài này sẽ lựa chọn tiếp cận theo phương pháp sử dụng bộ lọc Kalman để tích hợp dữ liệu của IMU và Lidar. Ngoài ra, Lidar cần nhiều thời gian hơn IMU để trả data trong một lần quét, các điểm đặc trưng có thể được lấy mẫu ở các thời điểm khác nhau dẫn đến chênh lệch không đồng đều giữa các mẫu do chuyển động của đối tượng trong khoảng thời gian này, vì vậy đề tài sẽ đồng bộ (khử lệch) dữ liệu Lidar [14] sử dụng phương pháp bù chuyển động [7].

Bên cạnh đó một số nghiên cứu còn kết hợp một số cảm biến khác như Encoder hoặc Camera, còn có GPS [8] để giúp tăng độ chính xác của bộ định vị. Nhóm tác giả trong [15] kết hợp dữ liệu vị trí 2D của Encoder và dữ liệu matching từ Camera, cùng với nhân tố Lidar odometry (có được dựa trên thuật toán LOAM [3]), tất cả đưa vào đồ thị nhân tố (factor graph) để tối ưu hóa vị trí. Còn nghiên cứu [16] lại kết hợp dữ liệu Encoder cùng với IMU trong mô hình toán tích phân đưa vào đồ thị nhân tố. [17] lại cung cấp phương pháp kết hợp Encoder và kết quả ước lượng IMU/Lidar dùng bộ lọc Kalman mở rộng nhưng cần phải đợi frame truyền Lidar kết thúc. Dựa trên thuật toán tích hợp IMU/Lidar ban đầu, luận văn cũng đưa ra phương pháp tích hợp IMU/Lidar với Encoder bằng cách sử dụng bộ lọc Kalman lặp kết hợp dữ liệu từ IMU và Encoder trong khoảng thời gian Lidar xử lý và gửi dữ liệu.

1.3. Nội dung và mục tiêu luận văn

1.3.1. Nội dung luận văn

- Xây dựng thuật toán định vị tích hợp IMU và Lidar sử dụng bộ lọc Kalman lặp. Từ đó phát triển thuật toán định vị tích hợp IMU, Lidar và Encoder giúp cải thiện độ chính xác vị trí.
- Xây dựng hệ thống phần cứng kết hợp IMU, Lidar, bên cạnh đó sử dụng RTK để làm tham chiếu vị trí; phần mềm cũng sẽ được xây dựng dựa trên nền tảng ROS dùng để thu thập dữ liệu từ cảm biến, xử lý tính toán giải thuật và hiển thị giám sát.
- Thực nghiệm thu thập dữ liệu và đánh giá hệ thống định vị.

1.3.2. Mục tiêu luận văn

- Sai số vị trí theo khoảng cách tại thời điểm cuối cùng không lớn hơn 5% tổng quãng đường di chuyển.
- Hệ thống định vị tích hợp đạt tần số cập nhật 10Hz.

1.4. Cấu trúc luận văn

Luận văn gồm có 6 chương với nội dung như sau:

- Chương 1: Giới thiệu tổng quan về đề tài.
- Chương 2: Trình bày tổng quan về cách thức hoạt động của các loại cảm biến, các cơ sở toán học liên quan đến quy ước trạng thái của hệ thống, lý thuyết Lie và bộ lọc Kalman lặp
- Chương 3: Trình bày về thuật toán tích hợp giữa IMU và Lidar. Từ đó phát triển giải thuật tích hợp IMU, Lidar và Encoder.
- Chương 4: Trình bày hệ thống phần cứng và phần mềm sử dụng cho thực nghiệm.
- Chương 5: Trình bày kết quả thực nghiệm và đánh giá sai số.
- Chương 6: Nêu ra kết luận và hướng phát triển của đề tài.

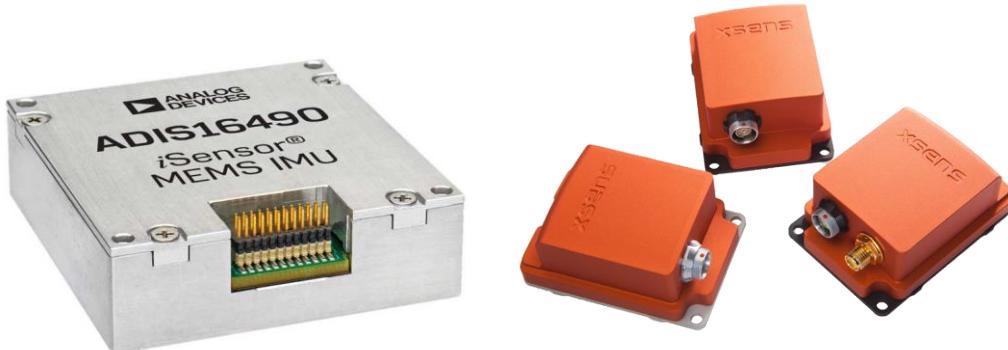
Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Nguyên lý hoạt động của các loại cảm biến

Như đã giới thiệu trong Chương 1, ba loại cảm biến chính được sử dụng trong luận văn bao gồm IMU, Lidar và Encoder. Thông tin từ cảm biến IMU sẽ được sử dụng cho bộ ước lượng quán tính, dữ liệu từ hai cảm biến còn lại (point cloud của Lidar và vận tốc có được từ Encoder) sẽ giúp cải thiện độ chính xác cho bộ ước lượng quán tính.

2.1.1. Cảm biến IMU

Cảm biến quán tính IMU (Inertial Measurement Unit) thông thường có 3 cảm biến chính là cảm biến vận tốc góc (gyroscope), cảm biến gia tốc (accelerometer) và cảm biến từ trường (magnetometer). Các cảm biến này đều cung cấp giá trị đo cho cả 3 trục. Ngoài ra, các IMU cũng thường tích hợp thêm cảm biến nhiệt độ hay áp suất để bù nhiễu. Trong phạm vi luận văn chỉ xét đến cảm biến gia tốc và cảm biến vận tốc góc. Các dòng sản phẩm IMU hiện nay thường được chế tạo bằng công nghệ MEMS hay còn được gọi là vi cơ điện tử. Những cảm biến dạng này có kích thước nhỏ gọn và dễ dàng lắp đặt trên các phương tiện tự hành, robot hoặc máy bay điều khiển từ xa. Một số hãng IMU nổi tiếng có thể nhắc đến như Analog Device, XSENS, ...

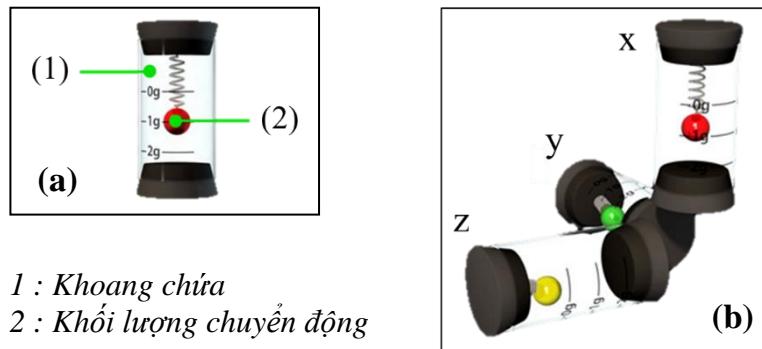


Hình 2.1. Một số sản phẩm IMU của hãng Analog Device và XSENS [18], [19]

2.1.1. Cảm biến gia tốc

Một mô hình đo gia tốc cơ bản bao gồm một khoang chứa hình trụ, bên trong có gắn lò xo, một đầu lò xo gắn với khoang chứa, đầu còn lại gắn với quả bóng hoặc vật nặng có khối lượng m như trong *Hình 2.2 (a)*. Khoang chứa này được gắn liền vào vật thể mà ta cần đo gia tốc, còn vật nặng (có thể là quả bóng) là vật có thể di chuyển một chiều bên trong khoang chứa. Vật nặng sẽ di chuyển bên trong khoang chứa khi khoang chứa bị dịch chuyển bởi chuyển động của vật thể, khiến lò xo dãn hoặc co ra [20]. Gọi k là độ cứng lò xo và x là độ dời của vật nặng trên trục Ox , áp dụng định luật Hooke và định luật 2 Newton, ta có phương trình cân bằng lực:

$$F_{spring} = -kx = ma_x \Rightarrow a_x = -\frac{kx}{m} \quad (2.1)$$

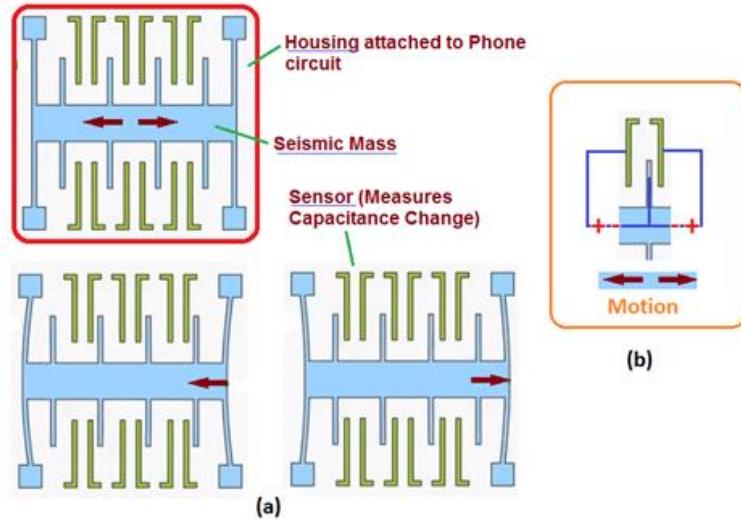


Hình 2.2. Mô hình đo gia tốc đơn trực (a) và mô hình đo gia tốc ba trực (b) [20]

Để đo được chuyển động của vật thể trong không gian ba chiều x , y , z , ta có thể sử dụng kết hợp ba mô hình đo gia tốc như trong *Hình 2.2 (b)*.

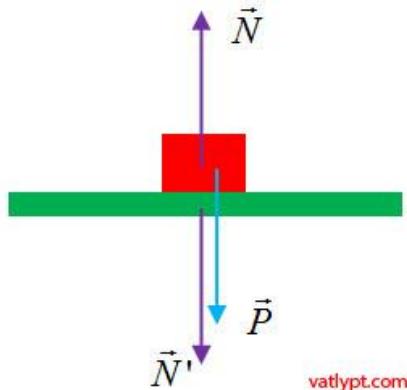
Cảm biến gia tốc MEMS không sử dụng hệ thống còng kènh như lò xo hay các vật nặng có khối lượng lớn, nhưng cũng hoạt động nguyên lý tương tự: Đo chuyển động của vật nặng bên trong khoang chứa và khoang chứa được gắn với vật cần đo gia tốc [20]. Vật nặng chuyển động trong cảm biến gia tốc MEMS chính là Seismic Mass như trong

Hình 2.3 (a). Khi khối silicon này chuyển động sẽ làm thay đổi điện dung của các tụ bán dẫn (*Hình 2.3 (b)*), sự thay đổi điện dung dẫn đến sự thay đổi về dòng điện, dựa vào sự thay đổi về dòng điện ta sẽ biết được sự thay đổi của gia tốc [21].



Hình 2.3. Mô hình đo gia tốc của cảm biến gia tốc dạng MEMS [21]

Cảm biến gia tốc có thể đo được gia tốc được tạo ra bởi các lực riêng biệt tác dụng vào vật thể (lực mà cảm biến cảm nhận được hay lực tiếp xúc (contact forces) trực tiếp với cảm biến) [22]. Khi vật thể đứng yên, lực mà cảm biến gia tốc cảm nhận được không phải là trọng lực vì trọng lực được sinh ra do ảnh hưởng của trọng trường Trái Đất. Nếu cảm biến nằm yên trên mặt phẳng ngang, lực nén lên mặt phẳng ngang gọi là áp lực N' có phương chiều như *Hình 2.4*, phản lực của mặt phẳng ngang tác dụng ngược lại vật một lực gọi là N . Về mặt độ lớn $N' = N =$ trọng lực P , N và N' là hai lực trực đối vì đặt vào hai vật khác nhau, nhưng N và P là hai lực cân bằng vì cùng tác dụng vào một vật [22]. Lực mà cảm biến cảm nhận được trong trường hợp này chính là N , mặt khác $\vec{N} = -\vec{P}$ nên gia tốc đo được là $-g$, với g là gia tốc trọng trường.



Hình 2.4. Minh họa về lực tác động lên cảm biến gia tốc khi nằm yên [23]

Ngoài ra khi vật thể chuyển động với gia tốc a_e , giá trị đo được từ cảm biến gia tốc a_m sẽ bao gồm cả $-g$ và gia tốc a_e gây ra bởi chuyển động. Dựa vào a_m và $-g$ ta sẽ tìm được a_e và từ a_e có thể suy ra được vận tốc và vị trí của vật nhờ các phương trình vi phân rời rạc sẽ được trình bày trong các phần sau.

$$a_m = a_e - g \quad (2.2)$$

$$a_e = a_m + g \quad (2.3)$$

Bên cạnh đó, giả sử phép đo gia tốc tồn tại độ lệch bias là b_a và nhiễu n_a , ta cần loại bỏ hai yếu tố lại khỏi phương trình (2.3).

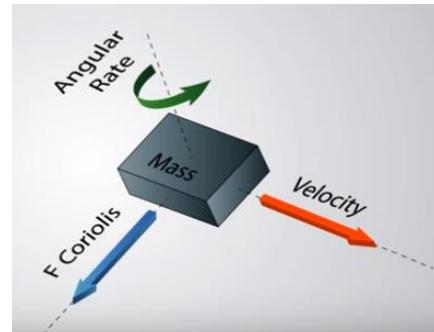
$$a_e = a_m - b_a - n_a + g \quad (2.4)$$

2.1.1.2. Cảm biến vận tốc góc

Về nguyên lý, cảm biến vận tốc góc MEMS hay Gyroscope xác định vận tốc góc xoay của vật thể trong không gian bằng cách sử dụng hiệu ứng Coriolis [20]. Hiệu ứng Coriolis là hiệu ứng xảy ra trong các hệ qui chiếu xoay so với các hệ quy chiếu quán tính [24]. Hiệu ứng Coriolis được minh họa như trong hình *Hình 2.5*: cho một vật có khối lượng m chuyển động với vector vận tốc v chuyển động trong hệ quy chiếu bất kỳ và hệ

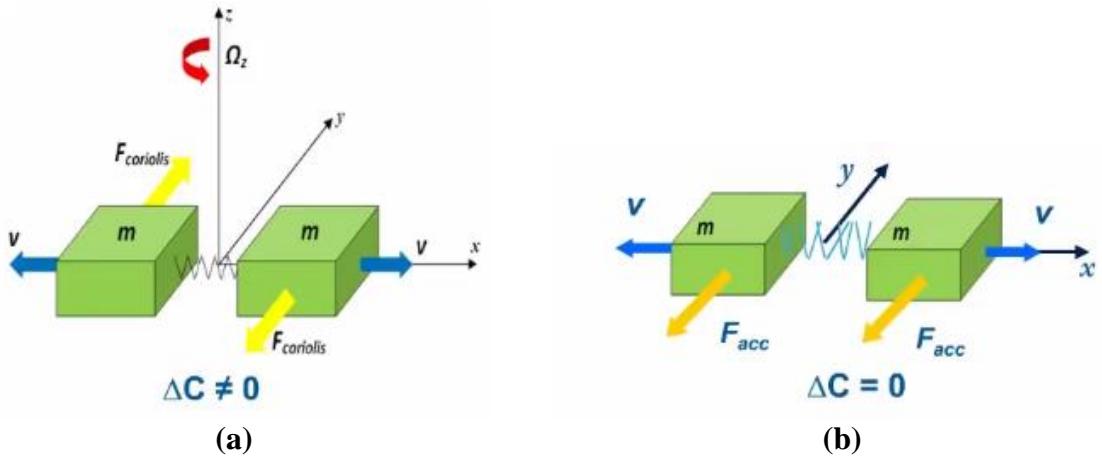
quy chiếu này xoay với vận tốc góc ω so với hệ quy chiếu quán tính. Khi đó, vật này chịu tác dụng của lực Coriolis F_c được xác định bởi công thức sau:

$$F_c = 2mv \times \omega \quad (2.5)$$



Hình 2.5. Minh họa về lực Coriolis [20]

Mô hình cảm biến vận tốc góc MEMS bao gồm hai khối silicon có kích thước như nhau và có cùng khối lượng m , hai khối silicon này được kích thích bởi một điện trường dao động khiến chúng cũng dao động và chuyển động liên tục ngược chiều nhau [25]. Khi hệ quy chiếu thực hiện phép xoay, hai lực Coriolis có cùng độ lớn nhưng ngược chiều nhau tác dụng vào hai khối silicon làm cho giá trị điện dung giữa chúng khác 0 ($\Delta C \neq 0$) như trong *Hình 2.6 (a)*. Trong khi đó hai khối silicon chịu tác dụng của hai lực cùng chiều khi chỉ có gia tốc chuyển động nên điện dung giữa chúng không thay đổi ($\Delta C = 0$). Nhờ cơ chế này cảm biến không bị ảnh hưởng do gia tốc chuyển động của hệ quy chiếu và có thể cảm nhận được sự thay đổi điện dung, từ đó đo được lực Coriolis. Bên cạnh đó khi biết được khối lượng m của vật và vận tốc chuyển động v của vật trong hệ quy chiếu đang xét, từ phương trình (2.5) ta có thể suy ra được vận tốc góc xoay.



Hình 2.6. Lực tác dụng lên cảm biến khi có vận tốc góc (a) và khi chỉ có gia tốc (b)

Dựa vào vận tốc góc có được ở trên, ta hoàn toàn có thể xác định các góc xoay của vật thể nếu biết được giá trị góc xoay ở thời điểm trước đó. Gọi ϕ, θ, ψ lần lượt là ba góc roll, pitch, yaw của vật thể như trong *Hình 2.7*. Ở thời điểm t_{k-1} , vận tốc góc đo được lần lượt là $\omega_{\phi,k-1}, \omega_{\theta,k-1}, \omega_{\psi,k-1}$, giả sử biết được giá trị ba góc roll, pitch, yaw tại thời điểm t_{k-1} là $\phi_{k-1}, \theta_{k-1}, \psi_{k-1}$, ta dễ dàng suy ra được ϕ, θ, ψ tại thời điểm t_k (chênh lệch thời gian giữa t_k và t_{k-1} là $\Delta t = t_k - t_{k-1}$) dựa vào các phương trình vi phân rời rạc.

$$\dot{\phi}_{k-1} = \frac{\phi_k - \phi_{k-1}}{\Delta t} = \omega_{\phi,k-1} \Rightarrow \phi_k = \phi_{k-1} + \omega_{\phi,k-1} \Delta t \quad (2.6)$$

$$\dot{\theta}_{k-1} = \frac{\theta_k - \theta_{k-1}}{\Delta t} = \omega_{\theta,k-1} \Rightarrow \theta_k = \theta_{k-1} + \omega_{\theta,k-1} \Delta t \quad (2.7)$$

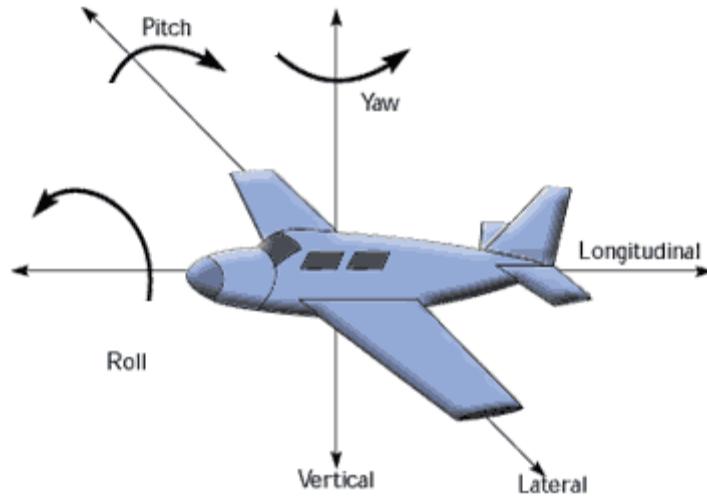
$$\dot{\psi}_{k-1} = \frac{\psi_k - \psi_{k-1}}{\Delta t} = \omega_{\psi,k-1} \Rightarrow \psi_k = \psi_{k-1} + \omega_{\psi,k-1} \Delta t \quad (2.8)$$

Tương tự phép đo gia tốc, phép đo vận tốc góc cũng tồn tại độ lệch bias là $b_\omega = [b_{\omega_\phi} \ b_{\omega_\theta} \ b_{\omega_\psi}]^T$ và nhiễu $n_\omega = [n_{\omega_\phi} \ n_{\omega_\theta} \ n_{\omega_\psi}]^T$.

$$\phi_k = \phi_{k-1} + (\omega_{\phi,k-1} - b_{\omega_{\phi,k-1}} - n_{\omega_{\phi,k-1}}) \Delta t \quad (2.9)$$

$$\theta_k = \theta_{k-1} + (\omega_{\theta,k-1} - b_{\omega_{\theta,k-1}} - n_{\omega_{\theta,k-1}}) \Delta t \quad (2.10)$$

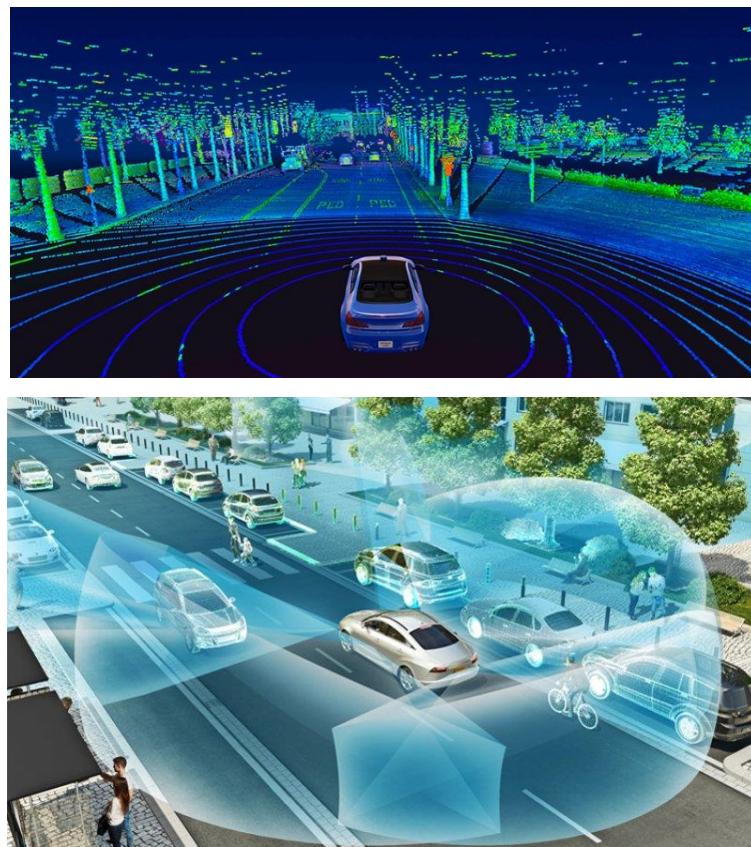
$$\psi_k = \psi_{k-1} + (\omega_{\psi,k-1} - b_{\omega\psi,k-1} - n_{\omega\psi,k-1}) \Delta t \quad (2.11)$$



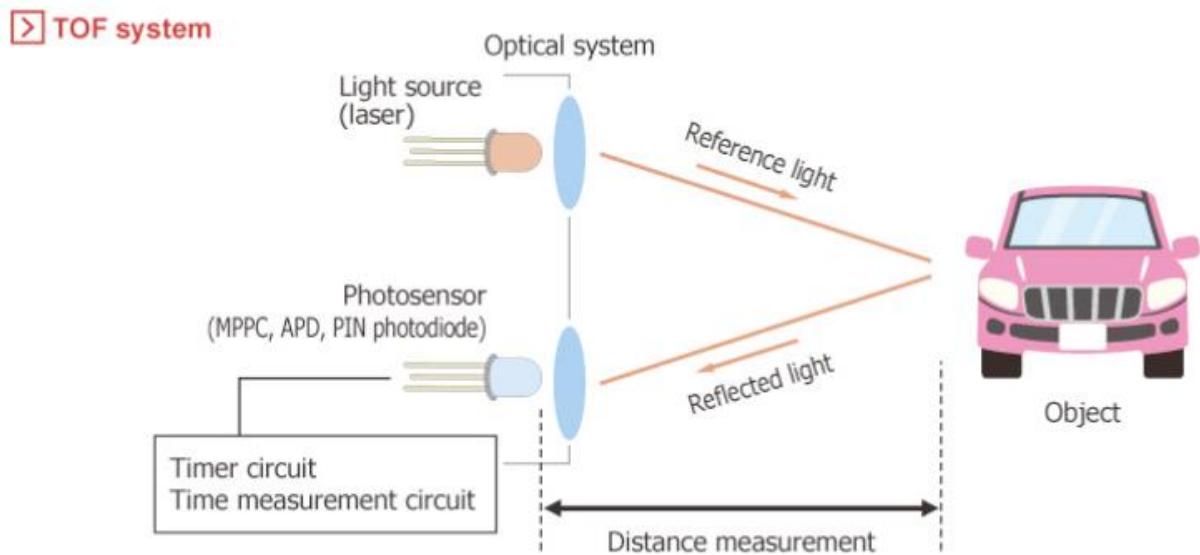
Hình 2.7. Minh họa ba góc roll, pitch, yaw của vật thể [26]

2.1.2. Cảm biến Lidar

Cảm biến LiDAR - Light Detection And Ranging là công nghệ sử dụng ánh sáng tia laser đo khoảng cách từ cảm biến đến vật thể bằng cách phát ra chùm tia laser có công suất thấp tới môi trường và nhận về chùm tia phản xạ để xử lý. Từ dữ liệu đó, hệ thống sẽ xây dựng bản đồ 3D mô phỏng không gian, đồ vật và con người [27]. Công nghệ LiDAR đóng một vai trò quan trọng trong việc phát triển các phương tiện tự lái. Với khả năng tạo bản đồ 3D về môi trường nhanh chóng, LiDAR giúp ô tô tự lái "nhìn thấy" xung quanh và cảnh báo các nguy hiểm tiềm ẩn khi tham gia giao thông [28].



Hình 2.8. Lidar được ứng dụng trong công nghệ ô tô [28], [29]



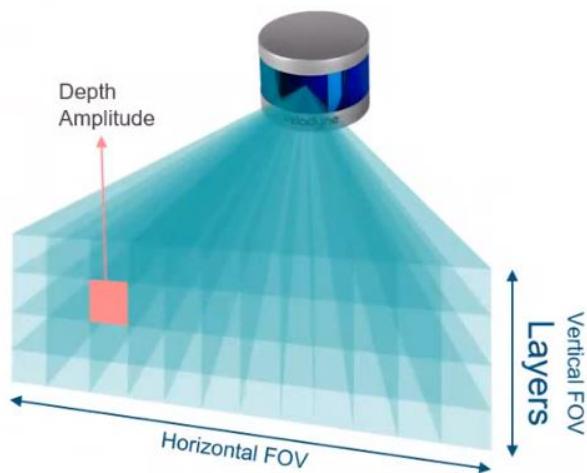
Hình 2.9. Minh họa hoạt động của cảm biến Lidar [30]

Lidar hoạt động theo cơ chế phát đi một chùm tia laser rồi thu nhận lại tín hiệu phản hồi. Tốc độ ánh sáng c² đã biết trước, độ trễ phản hồi t₁ được ghi nhận, từ đó tính được khoảng cách giữa máy phát và vật thể một cách tương đối chính xác.

$$d = ct/2 \quad (2.12)$$

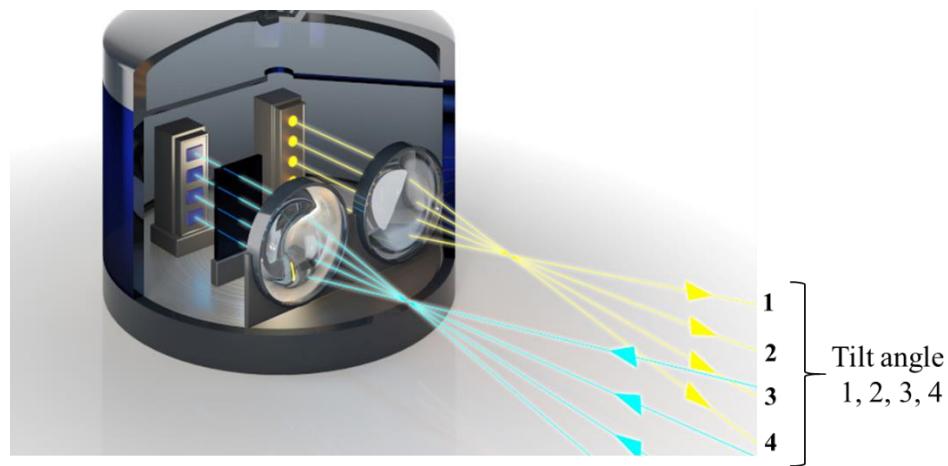
Lidar ghi nhận vật thể dưới dạng tọa độ số xyz. Đám mây điểm (xyz) này hợp thành mô hình 3 chiều. Trong một chu kỳ xử lý dữ liệu, Lidar liên tục phát tia laser đến các điểm khác nhau trên bề mặt đối tượng và nhận phản hồi, từ những thông tin khoảng cách khác nhau tại mỗi điểm trên đối tượng ta đó có thể dựng lên mô hình 3D của chúng. Ngoài thông tin khoảng cách, Lidar còn cung cấp dữ liệu về góc nghiêng cũng như là góc xoay của tia laser tại mỗi điểm giúp dễ dàng xác định các điểm này trong không gian 3 chiều. Tập hợp những điểm như vậy gọi mây điểm (Point Cloud). [31]

Một trong những thông số quan trọng của Lidar chính Field Of View (FOV) thể hiện giới hạn phát chùm tia laser và nhận chùm tia phản xạ, chỉ có những điểm nằm trong FOV mới được phát hiện bởi Lidar. FOV của Lidar được xác định bởi Depth Amplitude (biên độ sâu hay khoảng cách lớn nhất mà Lidar có thể nhận được tia phản xạ), Horizontal FOV (tầm quét ngang) và Vertical FOV (tầm quét dọc).

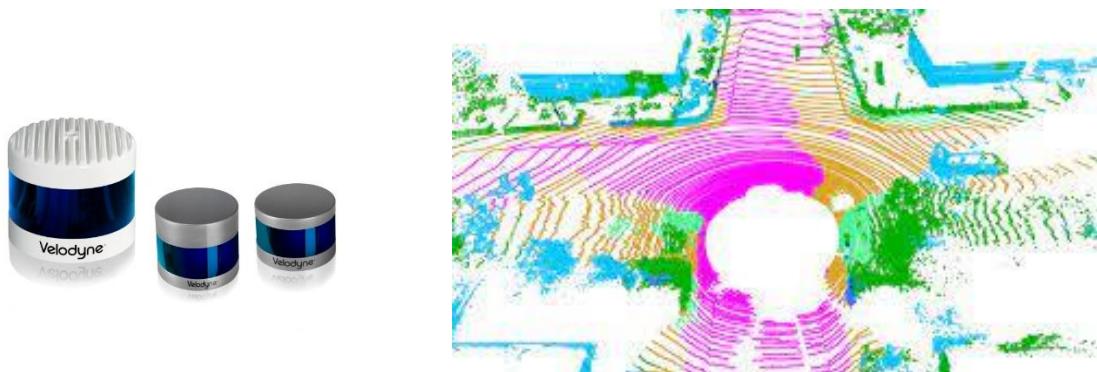


Hình 2.10. FOV của Lidar [32]

Các Lidar dạng cơ (Mechanical Lidar) truyền thống thường sử dụng nhiều bộ thu phát laser đặt chồng lên nhau để tăng Vertical FOV, mỗi bộ thu phát sẽ tương ứng với một góc nghiêng như trong *Hình 2.11*, do đó ta có thể suy ra Vertical FOV trong trường hợp này là $|tilt_angle_4 - tilt_angle_1|$ [33]. Các Lidar dạng này thường trang bị thêm một servo bên trong để có thể giúp các bộ thu phát laser xoay được 360 độ, từ đó làm tăng Horizontal FOV lên mức tối đa. Quan sát point cloud của dòng sản phẩm Lidar của hãng Velodyne trong *Hình 2.12*, ta thấy rằng đó là những vòng tròn đồng tâm, được gọi là ring, mỗi vòng tròn này tương ứng với tập hợp điểm quét được bởi một bộ thu phát laser.

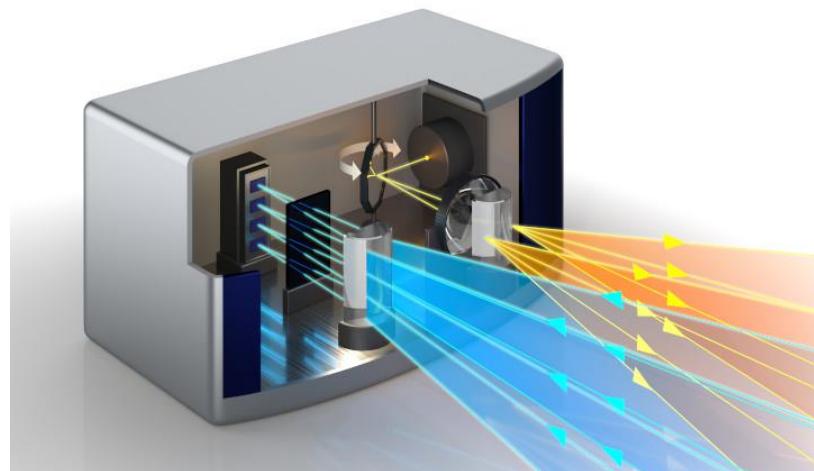


Hình 2.11. Cơ chế hoạt động của Lidar dạng cơ [34]

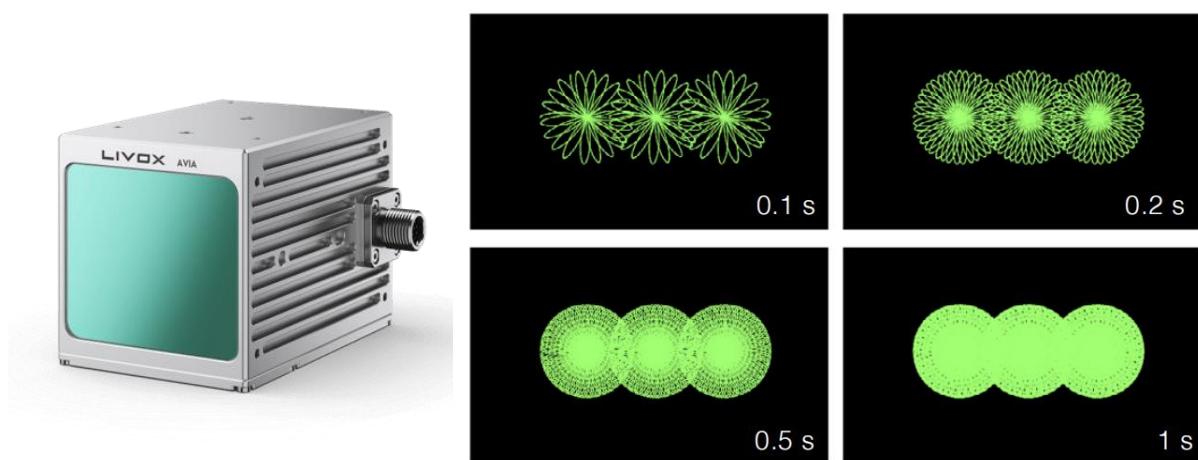


Hình 2.12. Các dòng sản phẩm Lidar cơ của Velodyne và hình dạng point cloud của nó [35], [36]

Ngoài ra trên thị trường hiện nay còn có các dòng sản phẩm Lidar dạng MEMS vi cơ điện tử hay còn có tên là Solid State Lidar. Các Lidar loại này chỉ sử dụng một đầu phát tia laser và sử dụng lăng kính để điều khiển hướng của tia laser (*Hình 2.13*), chính vì vậy chỉ cần điều khiển lăng kính này ta sẽ thu được những point cloud tùy ý theo mong muốn, chẳng hạn như point cloud hình cánh hoa của AVIA Lidar, một sản phẩm của hãng Livox (*Hình 2.14*). Nhưng hạn chế của AVIA Lidar là Horizontal FOV nhỏ nhưng ta hoàn toàn có thể trang bị thêm cho nó một cơ cấu chấp hành cơ khí gồm motor, encoder hoặc servo để giúp cảm biến xoay được 360 độ.



Hình 2.13. Cơ chế hoạt động của Lidar dạng MEMS [37]



Hình 2.14. AVIA Lidar của hãng Livox và hình dạng point cloud của nó [38], [39]

2.1.3. Cảm biến Encoder

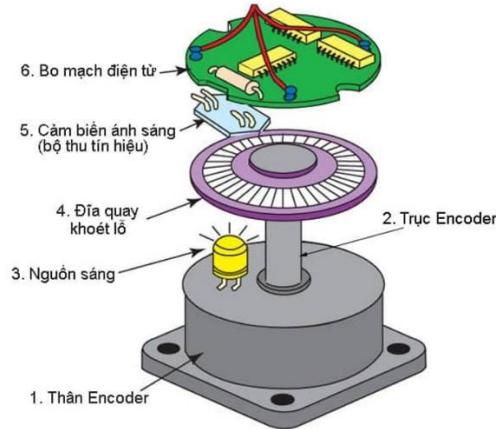
Encoder (Rotary Encoder) hay Bộ mã hóa vòng quay là một bộ cảm biến chuyển động cơ học tạo ra tín hiệu analog hoặc tín hiệu kỹ thuật số (digital) đáp ứng với chuyển động. Loại thiết bị cơ điện này có khả năng biến đổi chuyển động (chuyển động tịnh tiến, chuyển động quay của trục, ...) thành tín hiệu đầu ra số hoặc xung. Encoder được ứng dụng chủ yếu để phát hiện vị trí, hướng di chuyển, tốc độ... của động cơ bằng cách đếm số vòng mà trục quay được. Có thể liên tưởng rằng Encoder đóng vai trò như là bộ phận công tơ mét trên xe máy hay ô tô. Một số hãng Encoder có thể kể đến như Autonics hoặc Omron.



Hình 2.15. Encoder của hãng Autonics (a) và hãng Omron (b) [40], [41]

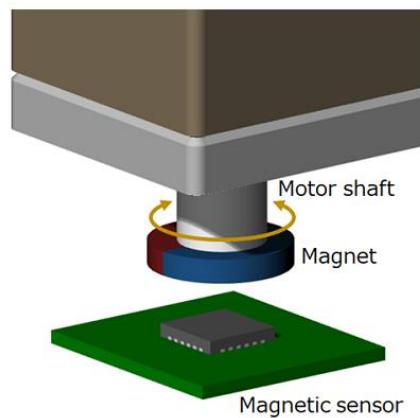
Các Encoder dạng quang (Opto Encoder) hoạt động theo nguyên lý đĩa quay quanh trục. Trên đĩa mã hóa có các rãnh nhỏ để nguồn phát sáng chiếu tín hiệu quang qua đĩa. Chỗ có rãnh thì ánh sáng xuyên qua được, chỗ không có rãnh ánh sáng không xuyên qua được. Với các tín hiệu có, hoặc không có ánh sáng chiếu qua, người ta ghi nhận được đèn led có chiếu qua lỗ hay không, từ đó cảm biến thu ánh sáng sẽ bật tắt liên tục và tạo

ra các xung vuông. Tín hiệu dạng xung này sẽ được truyền về bộ xử lý trung tâm (vi xử lý, PLC,...) để tính toán vị trí và tốc độ của động cơ. [42]



Hình 2.16. Nguyên lý hoạt động của Encoder quang [42]

Trong khi đó các Encoder từ (Magnetic Encoder) hoạt động dựa trên nguyên lý bao gồm một nam châm vĩnh cửu và một cảm biến từ. Nam châm vĩnh cửu được gắn vào trục động cơ, và một cảm biến từ được gắn cố định trên bảng mạch in (PCB) hoặc tương tự tại vị trí mà nó nhận được từ trường do nam châm vĩnh cửu tạo ra. Khi nam châm vĩnh cửu được gắn vào trục động cơ và quay theo, sự thay đổi và hướng của từ trường sẽ được cảm biến từ phát hiện, do đó có thể xác định vị trí quay và qua đó xác định được tốc độ quay của trục động cơ. [43]



Hình 2.17. Nguyên lý hoạt động của Encoder quang [43]

Tín hiệu đầu ra của Encoder thường có 2 loại là tín hiệu tuyệt đối và tín hiệu tương đối. Tín hiệu tuyệt đối cho ngõ ra dạng mã kỹ thuật số (BCD), Binary (nhị phân) hoặc Gray code, một giá trị tương ứng với một vị trí của encoder. Tín hiệu tương đối cho ngõ ra dạng xung vuông pha AB, hoặc ABZ hoặc ABZ đảo. Loại Encoder có 2 tín hiệu xung A và B là phổ biến nhất, tín hiệu khe Z là tín hiệu để xác định động cơ quay được một vòng. Đối với Encoder có ngõ ra là tín hiệu tương đối, giả sử tại thời điểm t_k , số xung đọc được là n_k , tại thời điểm t_{k+1} , số xung đọc được là n_{k+1} , $\Delta t = t_{k+1} - t_k$ là chênh lệch thời gian giữa hai thời điểm t_k và t_{k+1} , s là độ phân giải của Encoder và R là bán kính bánh xe thì vận tốc tại thời điểm k là:

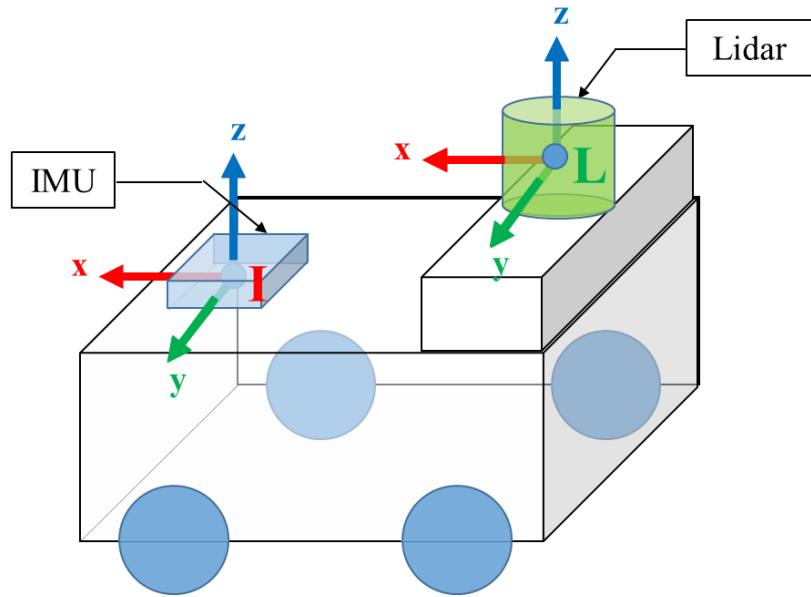
$$v_k = \frac{n_{k+1} - n_k}{\Delta t * s} 2\pi R \quad (2.13)$$

2.2. Các hệ tọa độ và các biến trạng thái của hệ thống

2.2.1. Các hệ tọa độ

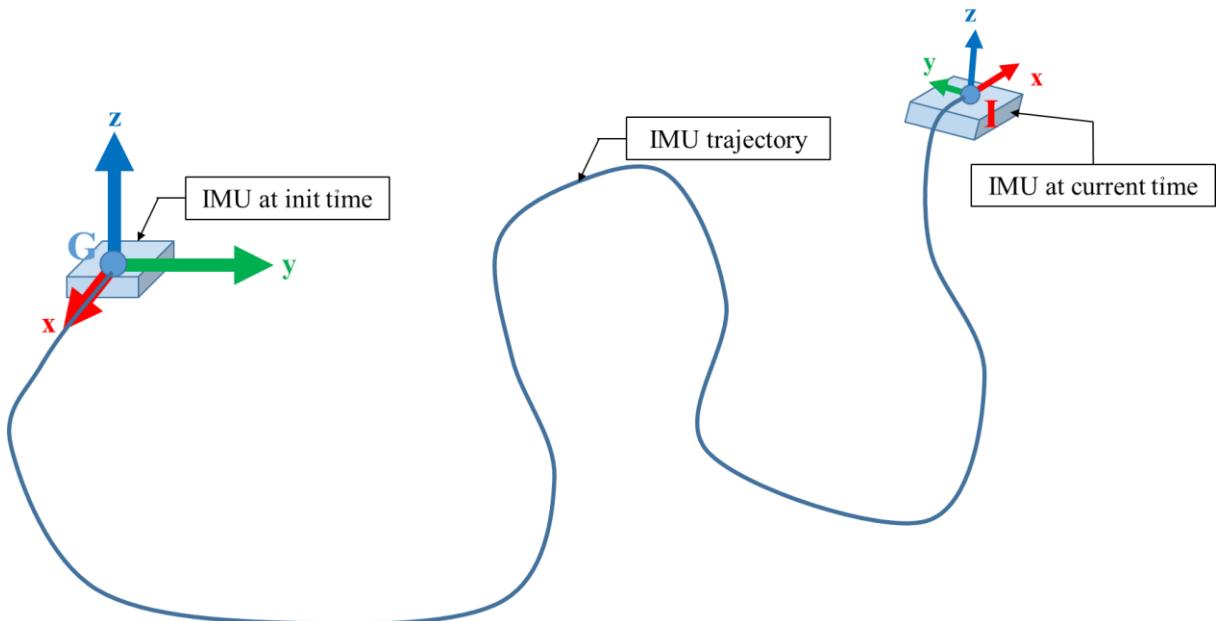
Ta đã tìm hiểu về các cảm biến trong phần 2.1, trong đó cần chú ý dữ liệu trả về từ IMU và Lidar là được tính trong hệ tọa độ gắn liền với cảm biến (*Hình 2.18*). Chính vì vậy ta cần quy đổi các dữ liệu từ hệ tọa độ IMU về hệ tọa độ Lidar hoặc từ hệ tọa độ Lidar về hệ tọa độ IMU. Trong phạm vi luận văn, các trạng thái ước lượng là của hệ tọa độ IMU so với hệ tọa độ Global nên ta sẽ chuyển đổi dữ liệu từ hệ tọa độ Lidar về hệ tọa độ IMU. Ký hiệu I là I-frame hay hệ tọa độ gắn với IMU, ký hiệu L là L-frame hay hệ tọa độ gắn với Lidar. Cả I-frame và L-frame đều có gốc tọa độ ở tâm cảm biến, chiều của trục x là chiều chuyển động, chiều của trục y là chiều hướng qua mạn trái của cảm biến, chiều của trục z là chiều hướng lên (*Hình 2.18*). Giả sử ${}^I R_L$ là ma trận xoay của L-frame đối với I-frame và ${}^I p_L$ là vị trí của L-frame trong I-frame, ta có ma trận biến đổi thuận nhất từ L-frame sang I-frame thường là:

$${}^I T_L = \begin{bmatrix} {}^I R_L & {}^I p_L \\ 0 & 1 \end{bmatrix} \quad (2.14)$$



Hình 2.18. Minh họa hệ tọa độ gắn liền với IMU và Lidar

Bên cạnh đó ký hiệu G là G-frame hay hệ tọa độ global hay hệ tọa độ toàn cục. Hệ tọa độ này thường được chọn tương ứng với hệ tọa độ gắn với IMU tại thời điểm đầu tiên, thời điểm mà tất cả các trạng thái vị trí, vận tốc, góc hướng đều bằng 0 (Hình 2.19).



Hình 2.19. Minh họa về hệ tọa độ Global và hệ tọa độ gắn liền với IMU

2.2.2. Các biến trạng thái

Các trạng thái được ước lượng bao gồm vị trí ${}^G p_I$, vận tốc ${}^G v_I$ và ma trận xoay ${}^G R_I$ của IMU trong hệ tọa độ global, bên cạnh đó là bias của cảm biến vận tốc góc b_g và bias của cảm biến gia tốc b_a . Ngoài ra ${}^G g$ là vector gia tốc trọng trường trong hệ global cũng được thêm vào bộ ước lượng trạng thái. Tất cả các thành phần ${}^G p_I$, ${}^G v_I$, b_g , b_a và ${}^G g$ đều là các vector 3×1 ngoại trừ ${}^G R_I$ là ma trận 3×3 .

$$x = [{}^G R_{I,3 \times 3}^T \quad {}^G p_{I,3 \times 1}^T \quad {}^G v_{I,3 \times 1}^T \quad b_{g,3 \times 1}^T \quad b_{a,3 \times 1}^T \quad {}^G g_{3 \times 1}]^T \quad (2.15)$$

Từ phương trình (2.2), có thể thấy chỉ cần để IMU đứng yên cảm biến gia tốc sẽ trả về $-g$ (phương trình (2.16)), vì vậy gia tốc trọng trường có thể được khởi tạo giá trị bằng cách để IMU đứng yên và đọc dữ liệu từ cảm biến gia tốc. Trong luận văn này, dữ liệu cảm biến gia tốc sẽ được thu thập trong một khoảng thời gian nhất định khi mà IMU không chuyển động (thường là khoảng thời gian lúc bắt đầu khảo sát), từ đó tính được độ lớn trung bình γ của vector gia tốc trong khoảng thời gian này, sau đó gia tốc trọng trường ${}^G g$ trong hệ global sẽ được khởi tạo ${}^G g = [0 \quad 0 \quad -\gamma]^T$ [7]. Vì giá trị đo từ cảm biến sẽ chịu ảnh hưởng của nhiễu và độ lệch bias nên ${}^G g$ cũng được đưa vào bộ ước lượng để được cập nhật cùng các biến trạng thái khác [13].

$$a_e = 0 \Rightarrow a_m = -g \quad (2.16)$$

Đặc biệt có thể thấy ma trận xoay ${}^G R_I$ sẽ được ước lượng thay vì ba góc roll, pitch, yaw được đề cập trong phương trình (2.9), (2.10), (2.11). Lý do ở đây là các phương trình vi phân quán tính không sử dụng trực tiếp roll, pitch, yaw mà thường sử dụng ma trận xoay ${}^G R_I$ có được từ ba góc roll, pitch, yaw nhưng về mặt toán học, phương trình vi phân của ${}^G R_I$ là tương đương với các phương trình (2.9), (2.10), (2.11) và từ ma trận xoay ${}^G R_I$, ta dễ dàng có được ba góc roll, pitch, yaw ϕ, θ, ψ nhờ phương trình (2.17), (2.18), (2.19) [42].

$$\phi = \arctan2\left({}^G R_I(3,2), {}^G R_I(3,3)\right) \quad (2.17)$$

$$\theta = -\arcsin\left({}^G R_I(3,1)\right) \quad (2.18)$$

$$\psi = \arctan2\left({}^G R_I(2,1), {}^G R_I(1,1)\right) \quad (2.19)$$

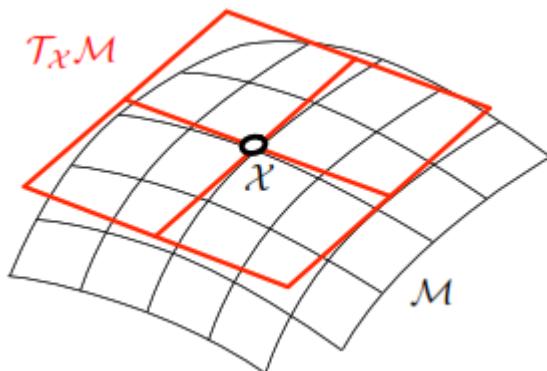
Việc tồn tại ma trận xoay ${}^G R_I$ kích thước 3×3 trong không gian trạng thái làm cho vector trạng thái trở nên phức tạp hơn. Vì vậy các phép toán thông thường đôi lúc sẽ trở nên vô nghĩa trong không gian trạng thái này. Do đó để thuật tiện tính toán, ta sẽ áp dụng lý thuyết Lie sẽ được trình bày trong phần sau.

2.3. Lý thuyết Lie

(Tài liệu tham khảo chính: [45])

2.3.1. Tổng quan

Trong toán học, một đa tạp trơn (smooth manifold) là một không gian topo, mỗi điểm trong không gian này có thể được xấp xỉ tuyến tính bằng một không gian vector. Ví dụ thường gặp nhất về đa tạp trơn chính là những siêu mặt phẳng, siêu mặt cong nhúng trong không gian đa chiều \mathbb{R}^n . Cho đa tạp trơn \mathcal{M} như *Hình 2.20*, luôn tồn tại duy nhất một không gian tiếp tuyến $T_X \mathcal{M}$ tại mỗi điểm X trong \mathcal{M} , cũng chính là xấp xỉ tuyến tính của \mathcal{M} tại X .



Hình 2.20. Minh họa đa tạp trơn \mathcal{M} và không gian tiếp tuyến $T_X \mathcal{M}$ của \mathcal{M} tại X [45]

Nhóm Lie (Lie group) là một đa tạp tron. Xét đa tạp tron \mathcal{M} là một nhóm Lie, nếu m là số bậc tự do của \mathcal{M} thì chênh lệch giữa các phần tử của \mathcal{M} có thể được biểu diễn thông qua các vector tuyến tính thuộc \mathbb{R}^m và ngược lại, độ dời giữa các vector thuộc \mathbb{R}^m cũng có thể được biểu diễn thông qua các phần tử của \mathcal{M} . Để phục vụ cho điều đó ta định nghĩa các phép toán \oplus và \ominus .

2.3.2. Phép \oplus và \ominus trong $SO(3)$

Trong các nhóm Lie, nhóm Lie ma trận (matrix Lie group) đóng vai trò quan trọng trong các bài toán robotics. Nhóm Lie ma trận bậc m là nhóm Lie mà các phần tử là các ma trận vuông khả nghịch $X \in \mathbb{R}^{m \times m}$. $SO(3)$ hay Special orthogonal group là một nhóm Lie ma trận. Để biểu diễn góc hướng của một hệ tọa độ trong không gian 3 chiều, ta sử dụng ma trận xoay $R \in SO(3)$. Chính vì vậy trong phạm vi luận văn chỉ xét đến phép toán \oplus và \ominus trong $SO(3)$.

$$\begin{aligned} \oplus : SO(3) \times \mathbb{R}^n &\rightarrow SO(3) & R \oplus r &= R \text{Exp}(r) \\ \ominus : SO(3) \times SO(3) &\rightarrow \mathbb{R}^n & R_1 \ominus R_2 &= \text{Log}(R_2^T R_1) \end{aligned} \quad (2.20)$$

$\text{Exp}(r)$ và $\text{Log}(R)$ được tính như sau:

$$\text{Exp}(r) = I + \frac{r}{\|r\|} \sin(\|r\|) + \frac{r^2}{\|r\|^2} (1 - \cos(\|r\|)) \quad (2.21)$$

$$\text{Log}(R) = \frac{\theta(R - R^T)^\vee}{2 \sin(\theta)} \quad (2.22)$$

Trong đó $\theta = \cos^{-1}\left(\frac{\text{trace}(R)-1}{2}\right)$, $R - R^T$ là một ma trận phản đối xứng có dạng

$$\begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}, \text{ từ đó suy ra } (R - R^T)^\vee = [\omega_x \quad \omega_y \quad \omega_z] \in \mathbb{R}^3.$$

Liên tưởng đến ma trận xoay ${}^G R_I$ trong mục 2.2.2, có thể tưởng tượng $R \in SO(3)$ chính là ${}^G R_{I,k}$ tại thời điểm t_k , $r \in \mathbb{R}^3$ là độ dời góc tại thời điểm t_{k+1} so với thời điểm t_k , ta dễ dàng tính được ${}^G R_{I,k+1} = {}^G R_{I,k} \oplus r$ chỉ bằng một phép tính mà không cần bước chuyển đổi trung gian nào cả.

2.3.3. Đạo hàm trong SO(3)

Ngoài ra ta cũng cần chú ý tới một số công thức tính đạo hàm trong $SO(3)$ để triển khai bộ lọc Kalman lặp trong không trạng thái ở mục 2.2.2. Với $Q, R \in SO(3)$ và $Q \ominus R = \theta \in \mathbb{R}^3$, ký hiệu $J_{\mathcal{X}}^y = \frac{Dy}{D\mathcal{X}}$, một số kết quả đạo hàm của phép \oplus và \ominus trong $SO(3)$ được rút ra như sau:

$$\begin{aligned} J_R^{R \oplus \theta} &= R(\theta)^T & J_{\theta}^{R \oplus \theta} &= J_r(\theta) \\ J_Q^{Q \ominus R} &= J_r^{-1}(\theta) & J_R^{Q \ominus R} &= -J_l^{-1}(\theta) \end{aligned} \quad (2.23)$$

Các đạo hàm này thỏa tính chất $J_l = J_r^T$ và $J_l^{-1} = J_r^{-T}$.

Công thức $J_l(\theta), J_l^{-1}(\theta)$ và $J_r(\theta), J_r^{-1}(\theta)$ được cho như sau:

$$J_l(\theta) = I + \frac{1 - \cos \theta}{\theta^2} [\theta]_{\Lambda} + \frac{\theta - \sin \theta}{\theta^3} [\theta]_{\Lambda}^2 \quad (2.24)$$

$$J_l^{-1}(\theta) = I - \frac{1}{2} [\theta]_{\Lambda} + \left(\frac{1}{\theta^2} - \frac{1 + \cos \theta}{2\theta \sin \theta} \right) [\theta]_{\Lambda}^2 \quad (2.25)$$

$$J_r(\theta) = I - \frac{1 - \cos \theta}{\theta^2} [\theta]_{\Lambda} + \frac{\theta - \sin \theta}{\theta^3} [\theta]_{\Lambda}^2 \quad (2.26)$$

$$J_r^{-1}(\theta) = I + \frac{1}{2} [\theta]_{\Lambda} + \left(\frac{1}{\theta^2} - \frac{1 + \cos \theta}{2\theta \sin \theta} \right) [\theta]_{\Lambda}^2 \quad (2.27)$$

Ngoài ra còn có một số công thức đạo hàm khác liên quan đến phép xoay vector $v \in \mathbb{R}^3$ và ma trận nghịch đảo của $R \in SO(3)$.

$$J_R^{Rv} = -R[v]_{\Lambda} \quad J_v^{Rv} = R \quad (2.28)$$

$$J_R^{R^{-1}} = -R \quad (2.29)$$

2.3.4. Đa tạp phức hợp các nhóm Lie

Gọi \mathcal{M} là một đa tạp phức hợp của các nhóm Lie, \mathcal{M} là hợp của M đa tạp thành phần không tương tác nhau. Gọi \mathcal{X} là một phần tử thuộc \mathcal{M} , \mathcal{X} là tập hợp các phần tử từ các đa tạp thành phần của \mathcal{M} .

$$\mathcal{M} = \langle \mathcal{M}_1 \quad \dots \quad \mathcal{M}_M \rangle \quad (2.30)$$

$$\mathcal{X} = \langle \mathcal{X}_1 \quad \dots \quad \mathcal{X}_M \rangle \in \mathcal{M}, \text{ với } \mathcal{X}_1 \in \mathcal{M}_1, \dots, \mathcal{X}_M \in \mathcal{M}_M \quad (2.31)$$

2.3.5. Phép \oplus và \ominus trong không gian trạng thái

Có thể thấy x trong công thức (2.15) là một phần tử thuộc đa tạp phức hợp $\mathcal{M} = \langle SO(3) \quad \mathbb{R}^{15} \rangle$, x có dạng $[R \quad a]^T$, trong đó $R \in SO(3)$ và $a \in \mathbb{R}^{15}$.

Phép \oplus và \ominus đối với thành phần $R \in SO(3)$ hoàn toàn tương tự như mục 2.3.2 đã trình bày. Phép \oplus và \ominus đối với thành phần a thuộc không gian vector \mathbb{R}^{15} chính là phép $+$ và $-$ vector thuần túy. Cụ thể với $R, R_1, R_2 \in SO(3)$ và $a, b, r \in \mathbb{R}^{15}$, ta có:

$$\begin{aligned} \begin{bmatrix} R \\ a \end{bmatrix} \oplus \begin{bmatrix} r \\ b \end{bmatrix} &= \begin{bmatrix} R \oplus r \\ a + b \end{bmatrix} \\ \begin{bmatrix} R \\ a \end{bmatrix} \ominus \begin{bmatrix} R_2 \\ b \end{bmatrix} &= \begin{bmatrix} R_1 \ominus R_2 \\ a - b \end{bmatrix} \end{aligned} \quad (2.32)$$

2.4. Bộ lọc Kalman lặp IEKF (Iterated Extended Kalman Filter) [44]

2.4.1. Giới thiệu

Rudolf Kalman (sinh năm 1930) là một nhà toán học Mỹ gốc Hungary là cha đẻ của phương pháp lọc Kalman (Kalman filter). Lọc Kalman là một thuật toán lọc nhiễu ra khỏi thông tin, có khả năng kết hợp nhiều tín hiệu với độ không chắc chắn khác nhau để cho ra kết quả ước lượng tốt nhất, trạng thái ước lượng hiện tại chỉ phụ thuộc vào trạng thái trước đó 1 chu kỳ tính toán nên tốc độ tính toán của bộ lọc Kalman khá nhanh nên được sử dụng rộng rãi trong nhiều lĩnh vực như điều khiển, hàng không, quân sự, vũ trụ, ngoài ra còn có nhận dạng tiếng nói, marketing v.v.

Tuy nhiên bộ lọc Kalman thông thường chỉ sử dụng hiệu quả đối với hệ thống tuyến tính trong khi hầu hết các hệ thống lại là phi tuyến. Vì vậy bộ lọc Kalman mở rộng (Extended Kalman Filter – EKF) sẽ được sử dụng trong trường hợp này. Bộ lọc EKF tuyến tính hóa hệ thống quanh giá trị ước lượng tại mỗi chu kỳ rồi áp dụng lọc Kalman lên mô hình xấp xỉ. Mặc dù vậy, do ảnh hưởng của yếu tố phi tuyến bộ lọc EKF không thể tối đa khả năng hậu nghiệm (Maximum a-Posteriori (MAP)) như bộ lọc Kalman trong hệ tuyến tính dẫn đến các điểm tuyến tính hóa sẽ ngày càng cách xa giá trị thực tế của nó [46]. Để khắc phục vấn đề này, chúng ta có thể sử dụng bộ lọc Kalman lặp (IEKF - Iterated Extended Kalman Filter) được xây dựng dựa trên thuật toán tối ưu Gauss – Newton [47].

Một bộ lọc Kalman sẽ hoạt động theo hai bước. Đầu tiên ước lượng trạng thái tại thời điểm t_k dựa vào mô hình toán của hệ thống và tín hiệu đầu vào ở thời điểm t_{k-1} . Tiếp theo đó các trạng thái được ước lượng ở bước đầu tiên sẽ được cập nhật với mô hình đo lường và giá trị đo từ cảm biến tại thời điểm t_k .

2.4.2. Mô hình toán hệ thống

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

$$\begin{cases} x_k = f_d(x_{k-1}, u_{k-1}, w_{k-1}) \\ y_k = h_d(x_k, n_k) \end{cases} \quad (2.33)$$

Trong đó f_d và h_d là các hàm phi tuyến biểu diễn mô hình quá trình và mô hình đo lường của hệ thống, x_k là vector trạng thái tại thời điểm k , u_{k-1} là tín hiệu đầu vào tại thời điểm $k-1$, y_k là tín hiệu ra được suy ra từ các biến trạng thái tại thời điểm k . w_{k-1} và n_k lần lượt là nhiễu trắng Gaussian của mô hình quá trình và mô hình đo với covariance tương ứng là Q_k và R_k .

2.4.3. Dự đoán trạng thái

Tương tự như bộ lọc EKF, bộ lọc IEKF cũng có bước dự đoán trạng thái tiên nghiệm (priori) \hat{x}_k và phương sai tiên nghiệm \hat{P}_k (phương trình (2.35) và (2.36)). Trong đó \bar{x}_{k-1} và \bar{P}_{k-1} là trạng thái và phương sai hậu nghiệm đã được cập nhật trong chu kỳ trước đó, còn F_{k-1} và G_{k-1} là các ma trận Jacobian được xác định bởi công thức (2.34).

$$F_{k-1} = \frac{\partial f_d}{\partial x_k} \Big|_{\bar{x}_{k-1}, u_{k-1}, 0} \quad G_{k-1} = \frac{\partial f_d}{\partial w_k} \Big|_{\bar{x}_{k-1}, u_{k-1}, 0} \quad (2.34)$$

$$\hat{x}_k = f_d(\bar{x}_{k-1}, u_{k-1}, 0) \quad (2.35)$$

$$\hat{P}_k = F_{k-1} \bar{P}_{k-1} F_{k-1}^T + G_{k-1} Q_{k-1} G_{k-1}^T \quad (2.36)$$

2.4.4. Cập nhật trạng thái

Đối với bộ lọc EKF, bài toán tối đa khả năng hậu nghiệm (posteriori) có thể xem như tương đương với cực tiểu hàm mục tiêu sau (2.37) [48], trong đó $\|x\|_\Sigma = x^T \Sigma x$ và y_k là giá trị đo lường. Lời giải tối ưu cho bộ lọc EKF được cho trong các phương trình (2.39), (2.40), (2.41), trong đó M_k và H_k là các ma trận Jacobian được xác định bởi công thức (2.38).

$$\min_{x_k} \|x_k - \hat{x}_k\|_{\hat{P}_k^{-1}} + \|y_k - h_d(x_k, 0)\|_{(M_k R_k M_k^T)^{-1}} \quad (2.37)$$

$$M_k = \frac{\partial h_d}{\partial n_k} \Big|_{\hat{x}_k, 0} \quad H_k = \frac{\partial h_d}{\partial x_k} \Big|_{\hat{x}_k, 0} \quad (2.38)$$

$$K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + M_k R_k M_k^T)^{-1} \quad (2.39)$$

$$\bar{x}_k = \hat{x}_k + K_k (y_k - h_d(\hat{x}_k, 0)) \quad (2.40)$$

$$\bar{P}_k = (I - K_k H_k) \hat{P}_k \quad (2.41)$$

Khác với bộ lọc EKF chỉ có một bước cập nhật duy nhất, bộ lọc IEKF đưa ra một kịch bản lặp ở bước cập nhật trạng thái nhằm đạt được kết quả ước lượng là cực đại khả năng hậu nghiệm. Tại mỗi lần lặp κ , IEKF sẽ tính toán giá trị cập nhật \hat{x}_k^κ và tuyến tính hóa hệ thống quanh \hat{x}_k^κ để phục vụ tính toán $\hat{x}_k^{\kappa+1}$ trong lần lặp tiếp theo. Quá trình này diễn ra cho đến khi $\|\hat{x}_k^{\kappa+1} - \hat{x}_k^\kappa\| < \epsilon$. Khi đó giá trị trạng thái cập nhật $\bar{x}_k = \hat{x}_k^{\kappa+1}$. Đặt $\tilde{x}_k^\kappa = x_k - \hat{x}_k^\kappa$, lúc này hàm mục tiêu cần cực tiểu sau khi tuyến tính hóa quanh \hat{x}_k^κ có dạng như công thức (2.42). Lúc này lời giải nghiệm tối ưu được đưa ra như trong phương trình (2.43) [46].

$$\begin{aligned} \min_{\tilde{x}_k^\kappa} & \|\hat{x}_k^\kappa - \hat{x}_k + \tilde{x}_k^\kappa\|_{\hat{P}_k^{-1}} \\ & + \|y_k - (h_d(\hat{x}_k^\kappa, 0) + H_k^\kappa \tilde{x}_k^\kappa)\|_{(M_k^\kappa R_k (M_k^\kappa)^T)^{-1}} \end{aligned} \quad (2.42)$$

$$\hat{x}_k^{\kappa+1} = \hat{x}_k + K_k^\kappa (y_k - h_d(\hat{x}_k^\kappa, 0) - H_k^\kappa (\hat{x}_k^\kappa - \hat{x}_k)) \quad (2.43)$$

Đặt $z_k^\kappa = h_d(\hat{x}_k^\kappa, 0)$, thực hiện một số phép biến đổi trên phương trình (2.43):

$$\begin{aligned} \hat{x}_k^{\kappa+1} &= \hat{x}_k + K_k^\kappa (y_k - z_k^\kappa - H_k^\kappa (\hat{x}_k - \hat{x}_k^\kappa)) \\ \Rightarrow \hat{x}_k^{\kappa+1} &= \hat{x}_k + K_k^\kappa (y_k - z_k^\kappa) - K_k^\kappa H_k^\kappa (\hat{x}_k - \hat{x}_k^\kappa) \\ \Rightarrow \hat{x}_k^{\kappa+1} &= \hat{x}_k - \hat{x}_k^\kappa + \hat{x}_k^\kappa + K_k^\kappa (y_k - z_k^\kappa) - K_k^\kappa H_k^\kappa (\hat{x}_k - \hat{x}_k^\kappa) \\ \Rightarrow \hat{x}_k^{\kappa+1} &= \hat{x}_k^\kappa + K_k^\kappa (y_k - z_k^\kappa) - (\hat{x}_k^\kappa - \hat{x}_k) + K_k^\kappa H_k^\kappa (\hat{x}_k^\kappa - \hat{x}_k) \end{aligned}$$

Cuối cùng ta được lời giải nghiệm tối ưu qua từng lần lặp như sau:

$$\hat{x}_k^{\kappa+1} = \hat{x}_k^\kappa + K_k^\kappa (y_k - z_k^\kappa) - (I - K_k^\kappa H_k^\kappa) (\hat{x}_k^\kappa - \hat{x}_k) \quad (2.44)$$

Áp dụng công thức (2.44) trong không gian phi tuyến của các nhóm Lie ta được phương trình (2.45), với $J_k^\kappa = \frac{\partial(x_k \ominus \hat{x}_k)}{\partial x_k} \Big|_{\hat{x}_k^\kappa}$ được thêm vào để tuyến tính hóa những thành phần phi tuyến trong không gian trạng thái (ví dụ như ma trận xoay). [46]

$$\hat{x}_k^{\kappa+1} = \hat{x}_k^\kappa \oplus K_k^\kappa (y_k - z_k^\kappa) - (I - K_k^\kappa H_k^\kappa) J_k^{\kappa-1} (\hat{x}_k^\kappa \ominus \hat{x}_k) \quad (2.45)$$

Bên cạnh đó tại mỗi lần lặp κ , các ma trận Jacobian và độ lợi Kalman sẽ được cập nhật lại với \hat{x}_k^κ :

$$H_k^\kappa = \frac{\partial h_d}{\partial x_k} \Big|_{\hat{x}_k^\kappa, 0} \quad M_k^\kappa = \frac{\partial h_d}{\partial n_k} \Big|_{\hat{x}_k^\kappa, 0} \quad J_k^\kappa = \frac{\partial(x_k \ominus \hat{x}_k)}{\partial x_k} \Big|_{\hat{x}_k^\kappa} \quad (2.46)$$

$$K_k^\kappa = J_k^{\kappa-1} \hat{P}_k J_k^{\kappa-T} H_k^{\kappa T} (H_k^\kappa J_k^{\kappa-1} \hat{P}_k J_k^{\kappa-T} H_k^{\kappa T} + M_k^\kappa R_k M_k^{\kappa T})^{-1} \quad (2.47)$$

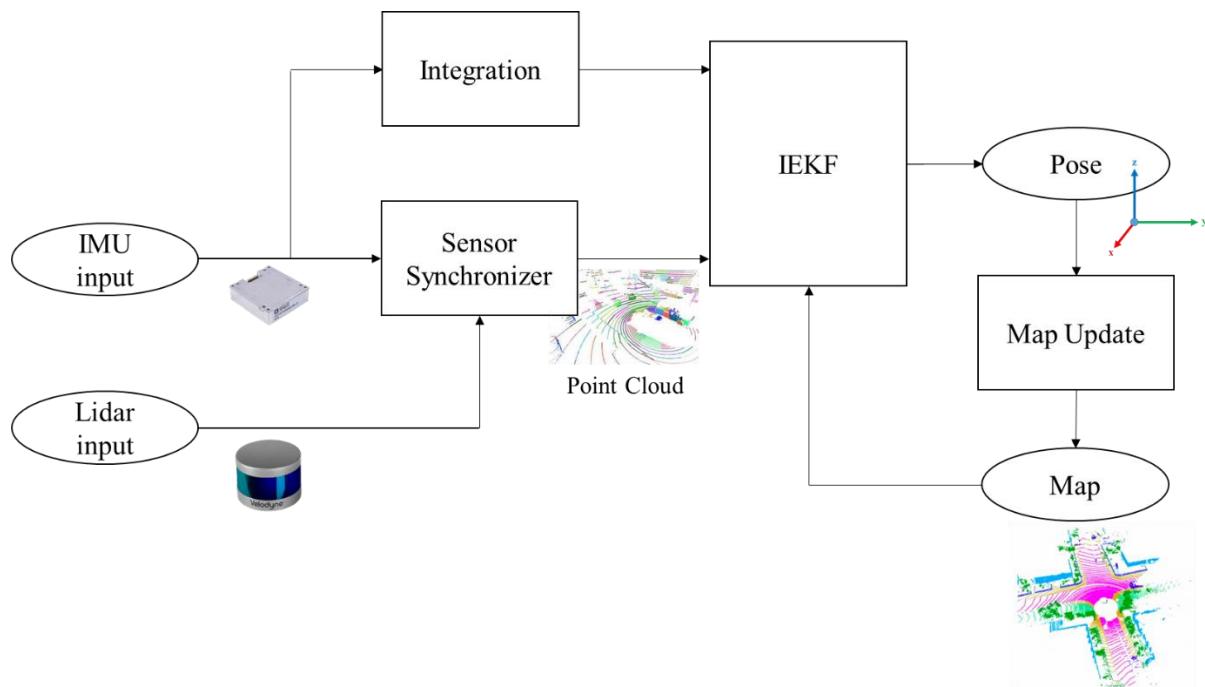
Phương sai hậu nghiệm sẽ được cập nhật một lần duy nhất sau khi $\hat{x}_k^{\kappa+1}$ hội tụ:

$$\bar{P}_k = (I - K_k^\kappa H_k^\kappa) (J_k^\kappa)^{-1} \hat{P}_k (J_k^\kappa)^{-T} \quad (2.48)$$

Chương 3. THUẬT TOÁN ĐỊNH VỊ TÍCH HỢP IMU/LIDAR

3.1. Tổng quan về thuật toán

IMU sẽ cung cấp dữ liệu gia tốc và vận tốc góc đưa vào khối tích phân Integration để ước lượng các giá trị vị trí, vận tốc, góc hướng trong khi Lidar đang xử lý dữ liệu cho một chùm tia quét. Sau khi Lidar hoàn thành tính toán và trả dữ liệu về, dữ liệu thô từ Lidar thu thập với tần số 10Hz được đồng bộ với dữ liệu IMU. Sau đó khoảng cách giữa các điểm trong point cloud với đường thẳng hoặc mặt phẳng tương ứng trong bản đồ hay nói cách khác là thặng dư (residual) được dùng để cập nhật trạng thái cho bộ ước lượng. Cuối cùng, các điểm trong mây điểm Lidar sẽ kết hợp với các trạng thái được cập nhật và được đưa vào bản đồ (Map).



Hình 3.1. Sơ đồ thuật toán định vị tích hợp IMU/Lidar

Integration block: Ước lượng các trạng thái dựa vào tích phân dữ liệu đo cảm biến vận tốc góc và cảm biến gia tốc trả về từ IMU sử dụng mô hình rời rạc của IMU.

Sensor Synchronizer: Các điểm trong point cloud có thể được lấy mẫu ở các thời điểm khác nhau trong một chu kỳ quét của Lidar dẫn đến sự không đồng bộ về thời gian, chênh lệch về chuyển động. Khối ***Sensor Synchronizer*** giúp hiệu chỉnh thời gian lấy mẫu của các điểm đặc trưng về cùng một thời điểm.

IEKF block: Các điểm đặc trưng trích xuất từ Lidar sau khi được hiệu chỉnh thời gian lấy mẫu được đưa vào bộ lọc IEKF để tính toán các thặng dư khoảng cách. Các thặng dư này được sử dụng để cập nhật các trạng thái ước lượng từ IMU. Quá trình tính toán thặng dư và cập nhật trạng thái sẽ được lặp đi lặp lại đến khi thỏa mãn điều kiện dừng. Chi tiết bộ lọc IEKF sẽ được trình bày trong các phần sau.

Map Update block: Sử dụng kết quả ước lượng chuyển point cloud về hệ tọa độ global và bổ sung point cloud này vào bản đồ.

Trong phạm vi luận văn sẽ tập trung chủ yếu vào 3 khối ***Integration***, ***Sensor Synchronizer*** và ***IEKF***.

3.2. Định vị quán tính sử dụng IMU

3.2.1. Mô hình phi tuyến IMU

Từ các mục 2.1.1.1 và 2.1.1.2, ta có được các phương trình quán tính của ma trận xoay, vị trí và vận tốc (phương trình (3.1), (3.2), (3.3)). Trong đó ${}^I\omega_m$ và Ia_m lần lượt là giá trị đo gia tốc và vận tốc góc của cảm biến IMU, các phép đo này chịu ảnh hưởng bởi hai đại lượng nhiễu trắng Gaussian n_g và n_a với phương sai nhiễu là q_g và q_a . Bên cạnh đó, các biến trạng thái b_g và b_a (bias của cảm biến vận tốc góc và cảm biến gia tốc của IMU) cũng sẽ được mô hình hóa bởi nhiễu trắng Gaussian n_{bg} và n_{ba} với phương sai nhiễu là q_{bg} và q_{ba} (phương trình (3.4), (3.5)). Cuối cùng, như đã đề cập ở mục 2.2.2, gia tốc trọng trường cũng sẽ được thêm vào không gian trạng thái, vì gia tốc trọng trường trong một khu vực nhất định được xem như không đổi nên ta có được phương trình (3.6).

$${}^G\dot{R}_I = {}^G R_I \left[{}^I\omega_m - b_g - n_g \right]_\wedge \quad (3.1)$$

$${}^G\dot{p}_I = {}^G v_I \quad (3.2)$$

$${}^G\dot{v}_I = {}^G R_I ({}^I a_m - b_a - n_a) + {}^G g \quad (3.3)$$

$$\dot{b}_g = n_{bg} \quad (3.4)$$

$$\dot{b}_a = n_{ba} \quad (3.5)$$

$${}^G\dot{g} = 0 \quad (3.6)$$

Với x được định nghĩ như trong công thức (2.15), $u = [{}^I\omega_m^T \quad {}^Ia_m^T]$ và $w = [n_g^T \quad n_a^T \quad n_{bg}^T \quad n_{ba}^T]^T$. Gọi f là hệ các phương trình trạng thái từ phương trình (3.1), (3.2), (3.3), (3.4), (3.5), (3.6), mô hình phi tuyến của IMU có thể được viết lại như phương trình (3.7). Rồi rạc phương trình (3.12) với khoảng thời gian lấy mẫu Δt , ta được mô hình rạc hóa của IMU (phương trình (3.8)).

$$\dot{x} = f(x, u, w) \quad (3.7)$$

$$x_{i+1} = x_i \oplus \Delta t f(x_i, u_i, w_i) \quad (3.8)$$

3.2.2. Tính toán trạng thái

Từ phương trình (3.8), với thời gian lấy mẫu $\Delta t = \tau_{i+1} - \tau_i$, τ_{i+1} và τ_i là hai thời điểm lấy mẫu liên tiếp của IMU. Ta sẽ sử dụng i ký hiệu cho chỉ số giá trị đo IMU trong khoảng thời gian một frame truyền của Lidar và k là ký hiệu cho chỉ số giá trị đo của Lidar. Gọi \hat{x}_i là giá trị các trạng thái tại thời điểm t_i , với $t_{k-1} < t_i < t_k$. Tại thời điểm bắt đầu chu kỳ quét của Lidar tức t_{k-1} thì $i = 0$, $\hat{x}_0 = \bar{x}_{k-1}$ với \bar{x}_{k-1} là trạng thái đã được cập nhật tại thời điểm kết thúc chùm tia quét Lidar trước đó. Trong khoảng thời gian từ t_{k-1} đến t_k (thời điểm kết thúc frame truyền Lidar), các biến trạng thái sẽ được dự đoán nhờ mô hình phi tuyến của IMU (phương trình (3.9)). Các trạng thái ước lượng từ IMU sẽ được sử dụng làm trạng thái dự đoán của bộ lọc Kalman lặp.

$$\hat{x}_{i+1} = \hat{x}_i \oplus \Delta t f(\hat{x}_i, u_i, 0) \quad (3.9)$$

3.2.3. Tính toán covariance (ma trận hiệp phương sai)

Bên cạnh đó phương sai tiên nghiệm cũng sẽ được tính toán đến khi kết thúc một chu kỳ quét của Lidar.

$$\hat{P}_{i+1} = F_i \hat{P}_i F_i^T + G_i Q_i G_i^T \quad (3.10)$$

Trong đó Q_i là ma trận phương sai nhiễu quá trình mô hình rời rạc [49]:

$$Q_i = F_i G_i Q G_i^T F_i^T \Delta t \quad (3.11)$$

Với Q là ma trận phương sai nhiễu mô hình liên tục:

$$Q = \begin{bmatrix} q_g & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & q_a & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & q_{bg} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & q_{ba} \end{bmatrix} \quad (3.12)$$

Đặt $f_d(x, u, w) = x \oplus \Delta t f(x, u, w)$. F_i và G_i được tính theo công thức:

$$F_i = \frac{\partial f_d}{\partial x_i} \Big|_{\hat{x}_i, u_i, 0}$$

$$= \begin{bmatrix} \text{Exp}(-(\omega_{m,i} - \hat{b}_{g,i})\Delta t) & 0 & 0 & -J_r((\omega_{m,i} - \hat{b}_{g,i})\Delta t)\Delta t & 0 & 0 \\ 0 & I & I\Delta t & 0 & 0 & 0 \\ -{}^G\hat{R}_{I_i}[a_{m,i} - \hat{b}_{a,i}] \Delta t & 0 & I & 0 & -{}^G\hat{R}_{I_i}\Delta t & I\Delta t \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (3.13)$$

$$\begin{aligned}
G_i &= \frac{\partial f_d}{\partial w_i} \Big|_{\hat{x}_i, u_i, 0} \\
&= \begin{bmatrix} -J_j^T ((\omega_{m,i} - \hat{b}_{g,i}) \Delta t) \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -{}^G\hat{R}_{I_i} \Delta t & 0 & 0 \\ 0 & 0 & I \Delta t & 0 \\ 0 & 0 & 0 & I \Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.14)
\end{aligned}$$

Trong đó J_r được định nghĩa như phương trình (2.26) mục 2.3 phần Cơ sở lý thuyết.

3.3. Cập nhật trạng thái từ dữ liệu của Lidar

3.3.1. Phương pháp đồng bộ dữ liệu cảm biến Lidar

Khi một chu kỳ quét của Lidar kết thúc, các điểm đặc trưng trích xuất được càn kết hợp với các trạng thái và phương sai đã được dự đoán bởi IMU cũng chính là \hat{x}_k và \hat{P}_k . Tuy nhiên thời điểm kết thúc một frame dữ liệu của Lidar là t_k trong khi các điểm đặc trưng có thể được lấy mẫu ở bất cứ thời điểm nào giữa t_{k-1} và t_k dẫn đến chênh lệch bất đồng bộ. Giả sử m là số điểm đặc trưng, mỗi điểm đặc trưng sẽ có thời điểm lấy mẫu là ρ_j và $\rho_j \in (t_{k-1}, t_k]$, bên cạnh đó ký hiệu ${}^{I_k}\check{x}_{I_j}$ là trạng thái tại thời điểm lấy mẫu điểm ρ_j so với thời điểm kết thúc frame truyền Lidar t_k . Để đưa các điểm đặc trưng này về thời điểm t_k , ta sử dụng phương pháp bù chuyển động bằng cách cho phương trình (3.15) vào vòng lặp đệ quy, j sẽ bắt đầu từ m , giá trị bắt đầu của các trạng thái được cho như trong (3.16).

$${}^{I_k}\check{x}_{I_{j-1}} = {}^{I_k}\check{x}_{I_j} \oplus \left(-\Delta t f \left({}^{I_k}\check{x}_{I_j}, u_j, 0 \right) \right) \quad (3.15)$$

$$\begin{aligned}
{}^{I_k}\check{R}_{I_m} &= I \\
{}^{I_k}\check{p}_{I_m} &= 0 \\
{}^{I_k}\check{v}_{I_m} &= {}^G\hat{R}_{I_k}^T {}^G\hat{v}_{I_k}
\end{aligned} \quad (3.16)$$

Khai triển phương trình (3.15) một cách chi tiết ta được:

$$\begin{cases} {}^{I_k} \check{R}_{I_{j-1}} = {}^{I_k} \check{R}_{I_j} \text{Exp} \left(-\Delta t \left({}^I \omega_{m,i-1} - \hat{b}_{g,k} \right) \right) \\ {}^{I_k} \check{p}_{I_{j-1}} = {}^{I_k} \check{p}_{I_j} - \Delta t {}^{I_k} \check{v}_{I_j} \\ {}^{I_k} \check{v}_{I_{j-1}} = {}^{I_k} \check{v}_{I_j} - {}^{I_k} \check{R}_{I_j} \Delta t \left({}^I a_{m,i-1} - \hat{b}_{a,k} \right) - \Delta t {}^G \hat{R}_{I_k}^T {}^G \hat{g}_k \end{cases} \quad (3.17)$$

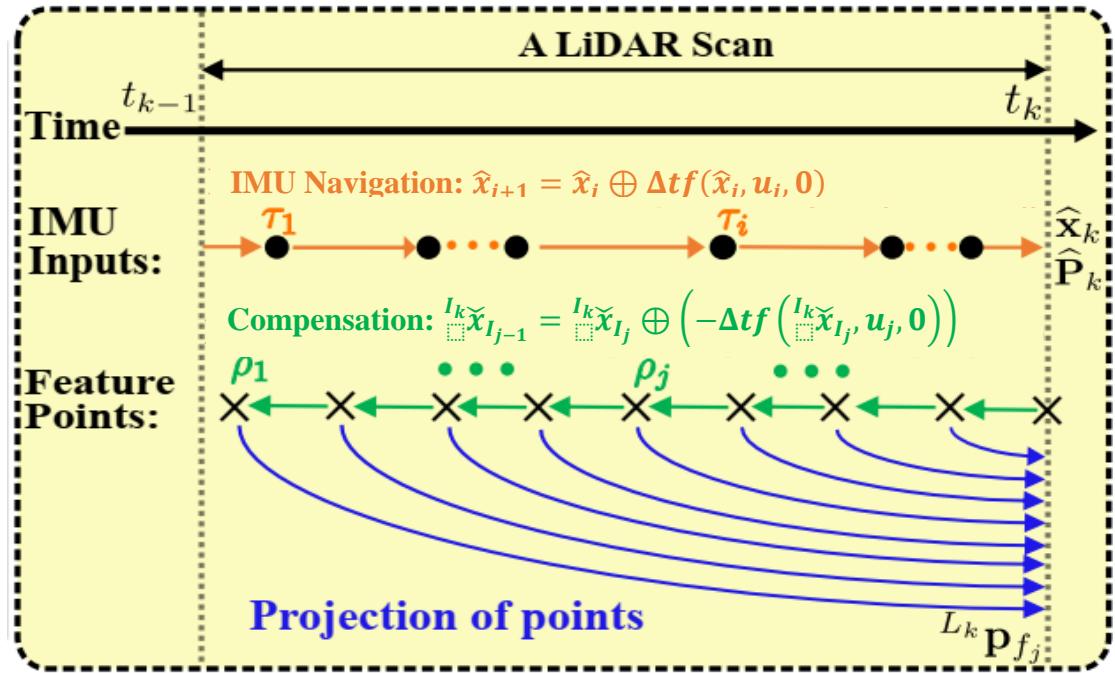
Vì $f \left({}^{I_k} \check{x}_{I_j}, u_j, 0 \right)$ bằng 0 đối với các thành phần bias cảm biến và gia tốc trọng trường nên ta sẽ sử dụng giá trị dự đoán của các biến trạng thái này là $\hat{b}_{g,k}$, $\hat{b}_{a,k}$ và ${}^G \hat{g}_k$ tại thời điểm t_k trong suốt quá trình tính toán bù chuyển động.

Ngoài ra Δt ở đây sẽ được tính bởi $\Delta t = \rho_j - \rho_{j-1}$, nếu thời điểm ρ_{j-1} nằm giữa hai thời điểm lấy mẫu của IMU là τ_{i-1} và τ_i ($\rho_{j-1} \in [\tau_{i-1}, \tau_i]$) thì giá trị đo tại thời điểm τ_{i-1} của IMU sẽ được sử dụng.

Sau khi toàn bộ $\left({}^{I_k} \check{R}_{I_j}, {}^{I_k} \check{p}_{I_j} \right)$ được tính, toàn bộ các điểm đặc trưng ${}^{L_j} p_{f_j}$ trong hệ tọa độ cục bộ L_j của Lidar sẽ được chuyển đổi về hệ tọa độ L_k của Lidar tại thời điểm kết thúc lần quét sử dụng ma trận biến đổi thuận nhất ${}^{I_k} \check{T}_{I_j} = \left({}^{I_k} \check{R}_{I_j}, {}^{I_k} \check{p}_{I_j} \right)$.

$${}^{L_k} p_{f_j} = {}^I T_L^{-1} {}^{I_k} \check{T}_{I_j} {}^I T_L {}^{L_j} p_{f_j} \quad (3.18)$$

${}^{L_k} p_{f_j}$ sẽ được sử dụng để tính toán thặng dư trong các phần sau.



Hình 3.2. Thời gian lấy mẫu IMU và các điểm trong một frame truyền Lidar [7]

3.3.2. Tính toán thặng dư

Các điểm đặc trưng trong mây điểm point cloud sau khi được hiệu chỉnh thời gian lấy mẫu về cùng một thời điểm là t_k cần được chuyển đổi về hệ tọa độ global để tính toán khoảng cách so với các cạnh và mặt phẳng tương ứng trong bản đồ global (phương trình (3.19)).

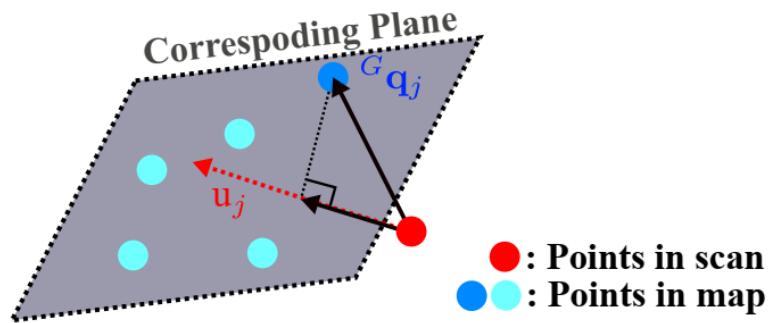
$${}^G p_{f_j} = {}^G \widehat{T}_{I_k} {}^I T_L {}^{L_k} p_{f_j} = {}^G \widehat{T}_{I_k} {}^{I_k} \check{T}_{I_j} {}^I T_L {}^{L_j} p_{f_j} \quad (3.19)$$

Mô hình đo lường của cảm biến Lidar chính là khoảng cách từ các điểm đặc trưng đến mặt phẳng hoặc đường thẳng mà điểm đó thực sự thuộc về trong bản đồ toàn cục (global map). Để thực sự tìm được đường thẳng hoặc mặt phẳng mà điểm đặc trưng thực sự thuộc về, ta sẽ lựa ra một số điểm lân cận điểm đó từ bản đồ, sau đó tìm ra mặt phẳng hoặc đường thẳng chứa các điểm lân cận đó, tiến hành tính khoảng cách từ điểm đặc trưng đến mặt phẳng hoặc đường thẳng đó, nếu khoảng cách nhỏ hơn một ngưỡng cho

trước, điểm đặc trưng xem như thuộc về mặt phẳng hoặc đường thẳng đó. Lúc này khoảng cách đó được gọi là thặng dư và được tính như trong công thức (3.20).

$$z_j(\hat{x}_k) = G_j \left({}^G p_{f_j} - {}^G q_j \right) = G_j \left({}^G \hat{T}_{I_k} {}^{I_k} \check{T}_{I_j} {}^I T_L {}^{L_j} p_{f_j} - {}^G q_j \right) \quad (3.20)$$

Đối với mặt phẳng $G_j = u_j^T$ với u_j là pháp vector của mặt phẳng. Đối với đường thẳng, $G_j = [u_j]_A$ với u_j là vector chỉ phương của đường thẳng. Còn ${}^G q_j$ là một điểm thuộc đường thẳng hoặc mặt phẳng đó. Vì trong môi trường thực tế, đặc trưng mặt phẳng thường dễ xác định hơn là đặc trưng đường thẳng nên phạm vi luận văn chỉ tập trung tìm mặt phẳng chứa điểm đặc trưng [50].



Hình 3.3. Minh họa tính toán thặng dư từ điểm đặc trưng của Lidar [50]

3.3.3. Mô hình đo lường

Giả sử giá trị đo ${}^{L_j} p_{f_j}$ trả về từ Lidar trong (3.20) chịu ảnh hưởng bởi nhiều, suy ra phép đo khoảng cách (3.20) sẽ chịu ảnh hưởng bởi nhiều đo lường $n_{f_j} \sim \mathcal{N}(0, R_{f_j})$, ta có được mô hình đo lường tại một điểm trong point cloud như phương trình (3.21).

$$h_j(x_k, {}^{L_j} n_{f_j}) = G_j \left({}^G \hat{T}_{I_k} {}^{I_k} \check{T}_{I_j} {}^I T_L {}^{L_j} p_{f_j} - {}^G q_j \right) + n_{f_j} \quad (3.21)$$

Từ đó suy ra mô hình đo lường của hệ thống dưới tác động của nhiễu $n_k = [n_{f_1} \ \cdots \ n_{f_m}]^T$, phương sai $R_k = diag(R_{f_1}, \dots, R_{f_m})$ với m là tổng số điểm đặc trưng của point cloud:

$$y_k = h(x_k, n_k) = \begin{bmatrix} h_1(x_k, n_1) \\ \vdots \\ h_m(x_k, n_m) \end{bmatrix} \quad (3.22)$$

Tại lần lặp thứ k của bộ lọc Kalman lặp, ta lần lượt xác định giá trị đo y_k , giá trị mô hình đo z_k^k , các ma trận Jacobian H_k^k, M_k^k, J_k^k như đã đề cập trong mục 2.4.4. Vì các điểm đặc trưng được xem như là thuộc các mặt phẳng tương ứng nên giá trị đo y_k có thể được xem là 0 (phương trình (3.23)), còn z_k^k (phương trình (3.24)) được xác định dựa trên phương trình (3.21) và (3.22). Các ma trận H_k^k, M_k^k, J_k^k được xác định như trong phương trình (3.25), (3.26), (3.27)

$$y_k = 0_{m \times 1} \quad (3.23)$$

$$z_k^k = [h_1(\hat{x}_k^k, 0) \ \cdots \ h_m(\hat{x}_k^k, 0)]^T \quad (3.24)$$

$$H_k^k = [H_1^{kT} \ \cdots \ H_m^{kT}]^T \quad (3.25)$$

$$M_k^k = [M_1^{kT} \ \cdots \ M_m^{kT}]^T \quad (3.26)$$

$$J^k = \frac{\partial(x_k \ominus \hat{x}_k)}{\partial x_k} \Big|_{\hat{x}_k^k} = \begin{bmatrix} J_r^{-1}(\hat{R}_{I_k}^k \ominus \hat{R}_{I_k}) & 0_{3 \times 15} \\ 0_{15 \times 3} & I_{15 \times 15} \end{bmatrix} \quad (3.27)$$

Trong đó, J_r^{-1} được xác định theo công thức (2.27), M_j^k trong (3.28) và H_j^k trong (3.30) được xác định theo công thức (2.46).

Tromg phương trình (3.28), hệ số nhỏ j ($j \in [1, m]$) đi kèm các giá trị của phần tử ma trận M_j^k chính là vị trí của phần tử đó trong ma trận M_k^k , từ đó suy ra M_k^k chính là ma trận đơn vị kích thước $m \times m$ (phương trình (3.29)).

$$M_j^\kappa = \left. \frac{\partial h_j}{\partial n_k} \right|_{\hat{x}_k^\kappa, 0} = [0_1 \quad \cdots \quad 1_j \quad \cdots \quad 0_m] \quad (3.28)$$

$$M_k^\kappa = I_{m \times m} \quad (3.29)$$

Và từ phương trình (3.30) kết hợp với công thức (2.28), ta có được ma trận H_j^κ trong phương trình (3.31).

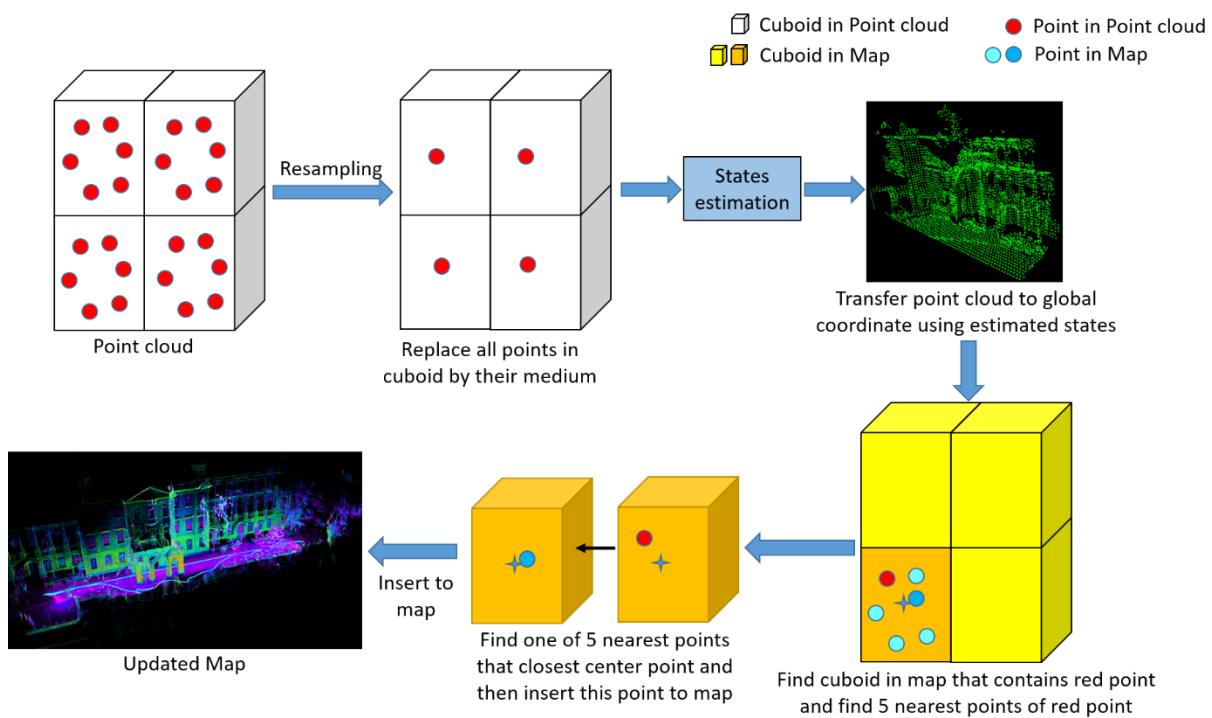
$$H_j^\kappa = \left. \frac{\partial h_j}{\partial x_k} \right|_{\hat{x}_k^\kappa, 0} = \left[\begin{array}{cccc} \left. \frac{\partial h_j}{\partial {}^G R_{I_k}} \right|_{\hat{x}_k^\kappa, 0} & \left. \frac{\partial h_j}{\partial {}^G p_{I_k}} \right|_{\hat{x}_k^\kappa, 0} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right] \quad (3.30)$$

$$H_j^\kappa = \left[-G_j {}^G R_{I_k} \left[{}^{I_k} \check{T}_{I_j} {}^I T_L {}^{L_j} p_{f_j} \right]_\Lambda \quad G_j \quad 0 \quad 0 \quad 0 \quad 0 \right] \quad (3.31)$$

Sau khi có giá trị đo y_k , giá trị mô hình đo lường z_k^κ và các ma trận Jacobian H_k^κ , M_k^κ , J_k^κ , dễ dàng áp dụng các công thức (2.45), (2.47), (2.48) để cập nhật các trạng thái và phương sai.

3.4. Cập nhật bản đồ

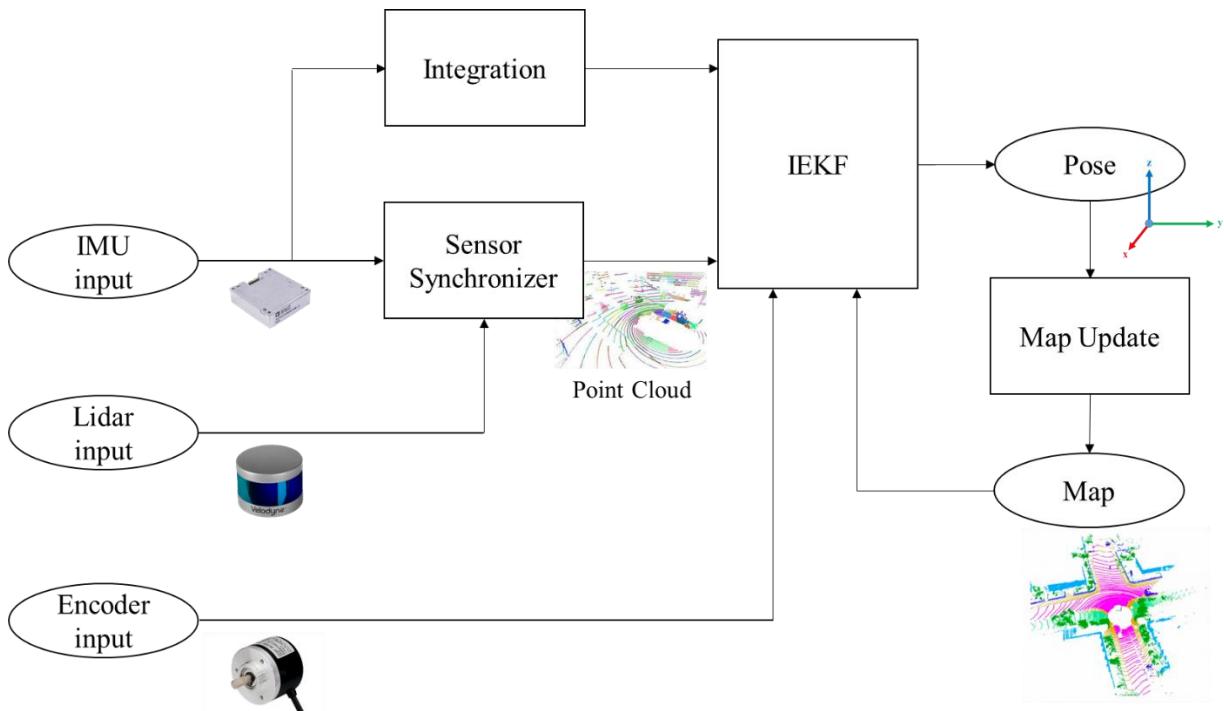
Point cloud trả về từ Lidar sẽ được lấy mẫu lại bằng cách chia point cloud thành những cuboid (kích thước 0.5m x 0.5m x 0.5m) và thay thế tất cả các điểm trong cuboid bằng trung bình cộng của chúng. Sau khi quá trình ước lượng trạng thái kết thúc, các điểm này sẽ được chuyển về hệ tọa độ global nhờ các trạng thái được cập nhật trước đó (phương trình (3.19)). Bản đồ lúc này cũng sẽ được chia thành các cuboid có kích thước giống nhau (0.5m x 0.5m x 0.5m), ta tiến hành đi tìm cuboid chứa điểm p (p nằm trong hệ global thuộc point cloud) và đi tìm 5 điểm trong bản đồ lân cận điểm p , tiếp theo trong 6 điểm (bao gồm p và các điểm lân cận) ta sẽ chọn ra một điểm gần trung tâm của cuboid nhất và thêm điểm đó vào bản đồ [51]. Quá trình cập nhật bản đồ được minh họa như trong Hình 3.4.



Hình 3.4. Minh họa quá trình cập nhật bản đồ

3.5. Thuật toán kết hợp Lidar/IMU/Encoder

Trong trường hợp có trang bị cảm biến Encoder, giá trị đo từ Encoder có thể được sử dụng cập nhật bộ ước trong quá trình chờ đợi dữ liệu từ Lidar, từ đó làm tăng độ chính xác khi ước lượng lượng vị trí.



Hình 3.5. Sơ đồ thuật toán định vị tích hợp IMU/Lidar

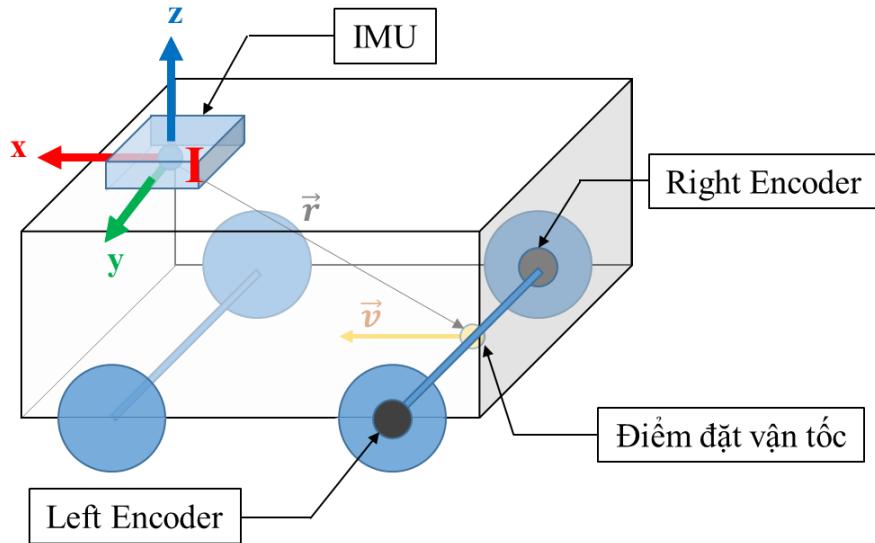
Giá trị đo vận tốc được tính từ Encoder như trong phương trình (3.32). Trong đó v_{left} và v_{right} lần lượt là tốc độ có được từ số xung Encoder của bánh sau bên trái và bánh sau bên phải (có được v_{left} và v_{right} nhờ vào công thức (2.13)).

$$y_k = \begin{bmatrix} \frac{v_{left} + v_{right}}{2} & 0 & 0 \end{bmatrix} \quad (3.32)$$

Trong tập dữ liệu Kaist [52], phương tiện di chuyển là xe hơi và có trang bị Encoder ở hai bánh sau, giả sử vận tốc tính được từ hai Encoder này chính là vận tốc tại điểm chính giữa trực nối hai bánh xe. Gọi \vec{r} là vector nối từ IMU đến điểm chính giữa trực nối bánh xe trong hệ tọa độ IMU hay hệ tọa độ body (Hình 3.6). Ta có mô hình đo vận tốc

trong hệ body như sau. Trong đó thành phần ${}^G R_I [{}^I \omega_k - b_g] \vec{r}$ có mục đích là bù lever arm offset do điểm đặt vận tốc không trùng với tâm của IMU [53]. Ngoài ra, phép đo này chịu ảnh hưởng bởi nhiễu trắng Gauss n_k với ma trận hiệp phương sai nhiễu là R_k , $n_k \sim \mathcal{N}(0, R_k)$

$$h(x_k, n_k) = {}^G R_I^T \left({}^G v_I - {}^G R_I [{}^I \omega_k - b_g] \vec{r} \right) + n_k \quad (3.33)$$



Hình 3.6. Minh họa hệ IMU và Encoder

Áp dụng công thức (2.46) và các công thức đạo hàm (2.28), (2.29) trong mục 2.3.3, ta tính được ma trận Jacobian H_k^κ (phương trình (3.34) và (3.35)).

$$\begin{aligned} H_k^\kappa &= \left. \frac{\partial h}{\partial x_k} \right|_{\hat{x}_k^\kappa, 0} \\ &= \left[\left. \frac{\partial h_j}{\partial {}^G R_{I_k}} \right|_{\hat{x}_k^\kappa, 0} \quad 0 \quad \left. \frac{\partial h_j}{\partial {}^G v_{I_k}} \right|_{\hat{x}_k^\kappa, 0} \quad \left. \frac{\partial h_j}{\partial b_{g_k}} \right|_{\hat{x}_k^\kappa, 0} \quad 0 \quad 0 \right] \end{aligned} \quad (3.34)$$

$$H_k^\kappa = \left. \frac{\partial h}{\partial x_k} \right|_{\hat{x}_k^\kappa, 0} = \left[{}^G R_I^T [{}^G v_I] {}^G R_I \quad 0 \quad {}^G R_I^T [{}^I \vec{r}] {}^G R_I \quad 0 \quad 0 \right] \quad (3.35)$$

Và tương tự phép đo của Lidar, ma trận Jacobian M_k^k trong phép đo của Encoder cũng là một ma trận đơn vị (phương trình (3.36)).

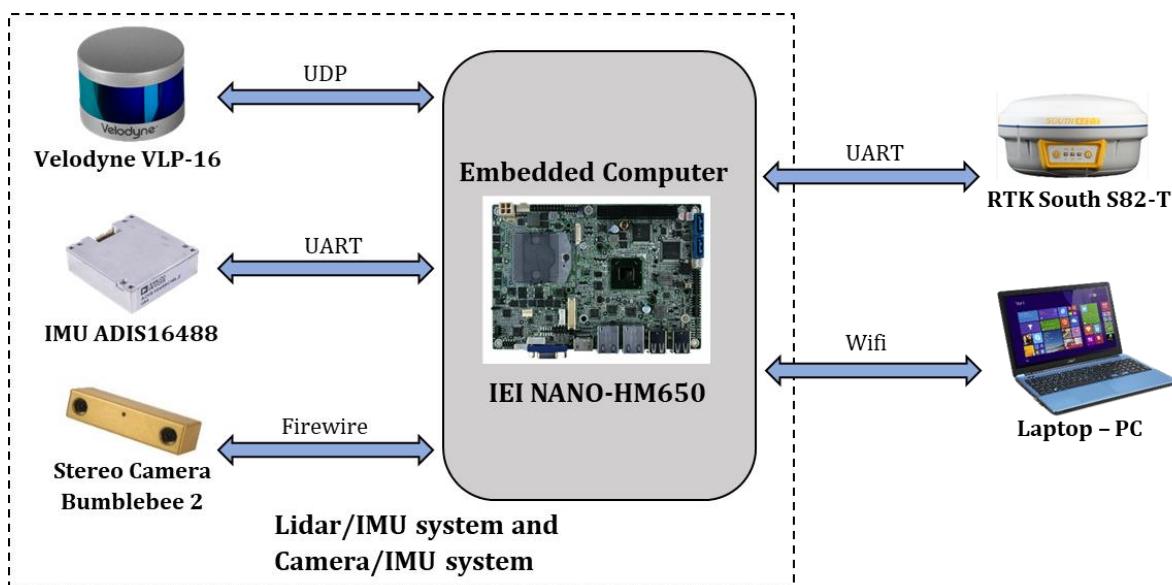
$$M_k^k = I_{3 \times 3} \quad (3.36)$$

Sau khi có đầy đủ giá trị đo từ cảm biến y_k , giá trị mô hình đo z_k^k và các ma trận Jacobian H_k^k, M_k^k, J_k^k (được xác định trong (3.27)), dễ dàng áp dụng các công thức (2.45), (2.47), (2.48) để cập nhật các trạng thái và phương sai.

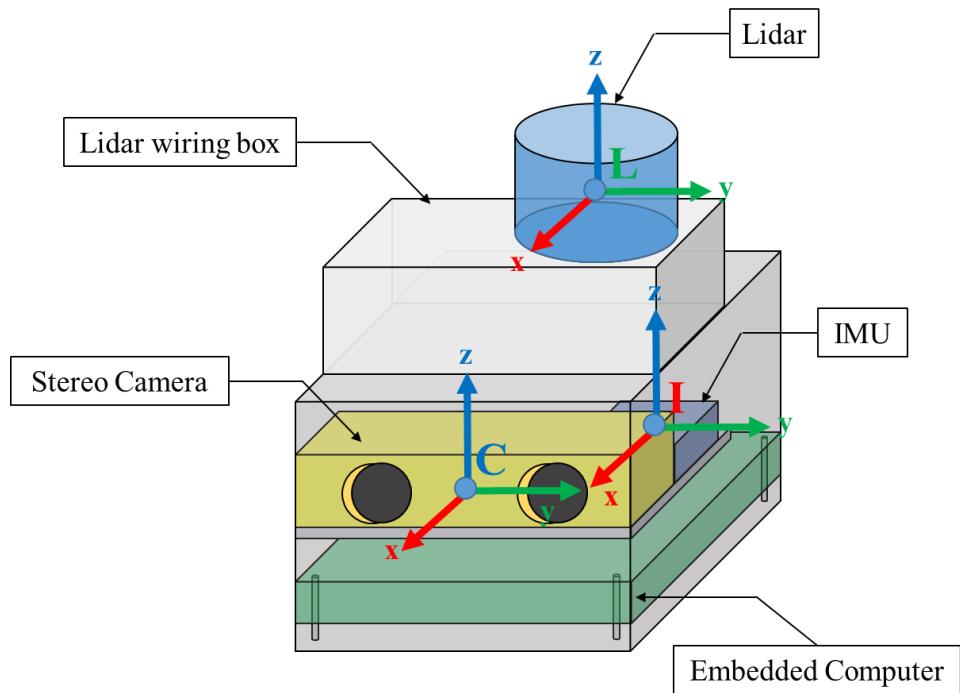
Chương 4. HỆ THỐNG PHẦN CỨNG VÀ PHẦN MỀM

4.1. Cấu trúc phần cứng

Trong đề tài, bên cạnh hệ định vị tích hợp IMU/Lidar, kết quả ước lượng vị trí từ hệ định vị tích hợp IMU/Camera [54] cũng sẽ được thu thập để so sánh với kết quả với hệ IMU/Lidar. Chính vì vậy dữ liệu từ Stereo Camera cũng sẽ được thu thập cùng với dữ liệu từ Lidar và IMU. Ngoài ra, RTK S82-T của hãng South Group sẽ được sử dụng làm tham chiếu vị trí để tính sai số vị trí ước lượng từ hệ IMU/Lidar và hệ IMU/Camera, từ đó làm cơ sở so sánh kết quả ước lượng vị trí của hai hệ thống tích hợp. Các thiết bị được kết nối với nhau và bố trí lắp đặt theo sơ đồ bên dưới. Trong đó một laptop được dùng để điều khiển, giám sát hoạt động của máy tính nhúng và hiển thị kết quả thông qua wifi.



Hình 4.1. Sơ đồ kết nối phần cứng hệ thống thực nghiệm



Hình 4.2. Mô hình phần cứng hệ thống thực nghiệm

IMU ADIS16488 là một cảm biến quán tính của ANALOG DEVICES có thể đo được ba trục gia tốc tịnh tiến, vận tốc góc và từ trường. ADIS16488 được chế tạo dưới dạng cảm biến vi cơ điện tử (MEMS) nên có kích thước nhỏ gọn dễ dàng tích hợp trong các board mạch. ADIS16488 hoạt động ở tầm nhiệt độ từ -55°C đến 105°C , tần số tối đa lên đến 500Hz. Nguồn sử dụng cấp cho ADIS16488 từ 3.0V đến 3.6V. [55]

IMU được đọc bởi một mạch xử lý sử dụng vi điều khiển STM32F4 thông qua giao thức SPI. Mạch xử lý giúp chuyển đổi sang chuẩn giao tiếp UART và phát ra với tần số tối đa 500Hz.



Hình 4.3. Cảm biến IMU ADIS16488 [56]

Cảm biến Lidar được sử dụng trong luận văn là Velodyne VLP-16 của hãng Velodyne Lidar. Đây là cảm biến Lidar 3D có tầm quét lên đến 100m, FoV (Field of View) 360° theo chiều ngang và 30° theo chiều dọc. VLP-16 có thể trả về dữ liệu lên đến 300000 điểm/s với độ chính xác thông thường rơi vào khoảng $\pm 3\text{ cm}$. Thiết bị hoạt động với điện áp 9-18 VDC, tầm nhiệt độ từ -10°C đến 60°C . Lidar Velodyne VLP-16 giao tiếp qua chuẩn UDP, tốc độ có thể lên đến 100 Mbps. [57]



Hình 4.4. Cảm biến Lidar Velodyne VLP-16 [29]

Stereo camera Point Grey Bumblebee2 [58]: camera gồm hai camera có baseline là 12cm, mỗi camera của Bumblebee2 có góc nhìn 97° và có tiêu cự 2.5mm, dùng để chụp đồng bộ cặp ảnh xám, kích thước ảnh 640x480 pixel, tần số cập nhật của camera là 20Hz, chuẩn giao tiếp FireWire (IEEE 1394a).



Hình 4.5. Stereo camera Point Grey Bumblebee2 [59]

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



Hình 4.6. Máy tính nhúng IEI NANO-HM650

GPS RTK South S82-T: GPS RTK của hãng South, sai số ở chế độ Fixed RTK là 1cm theo phương ngang và 2cm theo phương dọc, nhiệt độ vận hành từ -25°C đến 60°C, được dùng làm vị trí tham chiếu để đánh giá hệ thống thử nghiệm ngoài trời. [60]



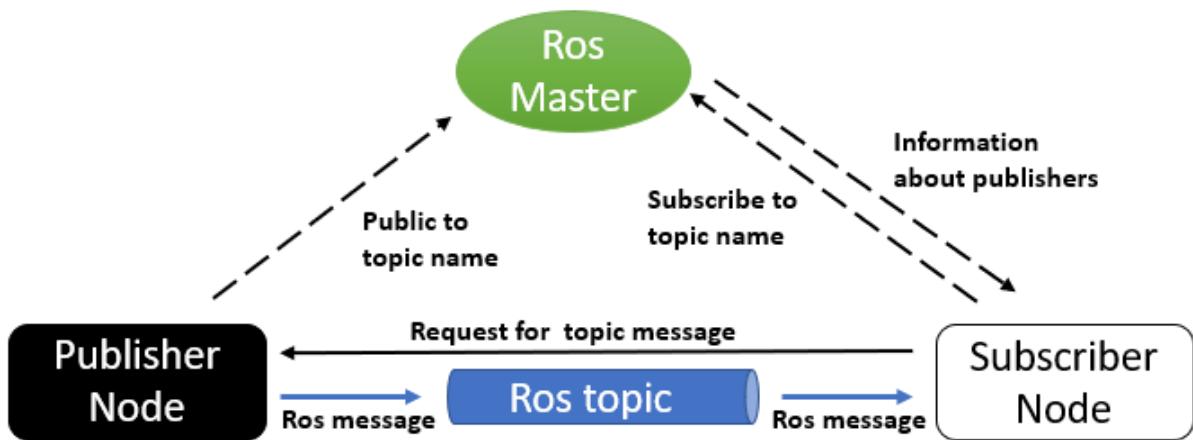
Hình 4.7. GPS RTK South S82-T

4.2. Cấu trúc phần mềm

Toàn bộ phần mềm trong phần này sẽ được tích hợp vào máy tính nhúng (Hệ điều hành Ubuntu 16.04), sử dụng ngôn ngữ chủ yếu là C++. Ngoài ra hệ thống phần mềm được xây dựng trên nền tảng ROS giúp dễ dàng kế thừa được các tính năng và thư viện sẵn có.

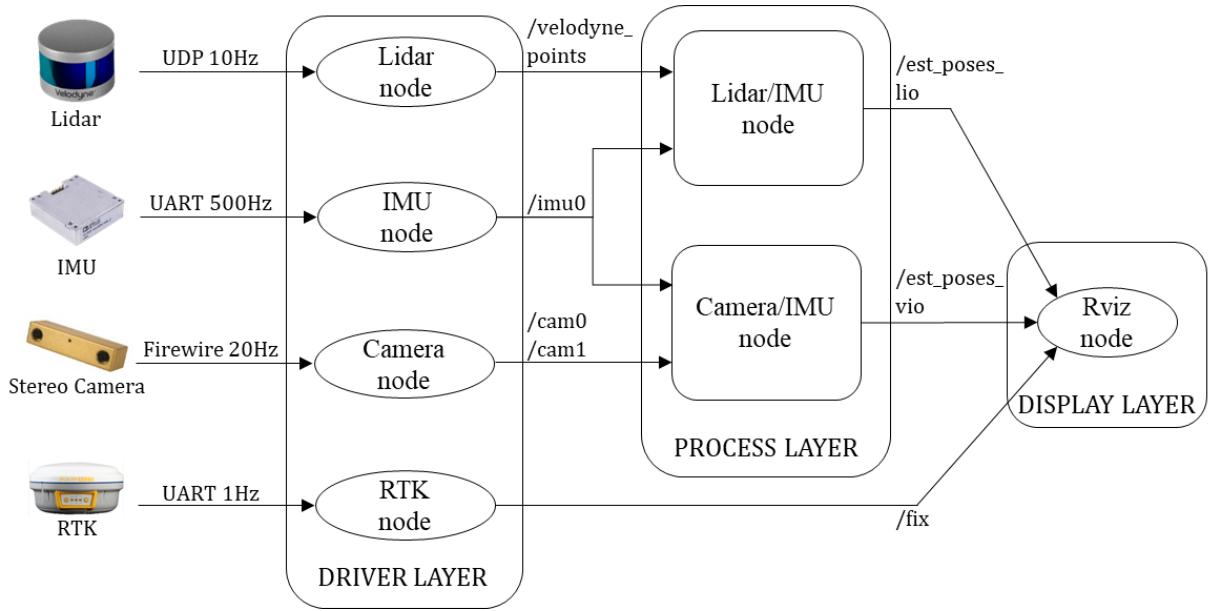
ROS hay Robot Operating System là một nền tảng mã nguồn mở, chuyên dụng thường được tích hợp cho robot, là một “robot frameworks”. Trong ROS, Ros Master sẽ là một node trung gian nhận đăng ký từ Publisher và Subscriber. Bất kì từ Publisher hoặc Subscriber nào muốn gửi data hoặc nhận data đều phải đăng ký (register) với Ros Master các thông tin về Topic và kiểu Message với Message là kiểu thông tin hay kiểu dữ liệu muốn gửi hoặc nhận, còn Topic có thể xem như tên định danh của Message đó. Ví dụ như thông tin từ cảm biến A và thông tin từ cảm biến B cùng kiểu dữ liệu với nhau hay

chung kiểu message nhưng là hai thông tin riêng biệt, vì vậy mà ta có thể định danh hai thông tin này bằng cách đặt tên cho chúng thông qua Topic.

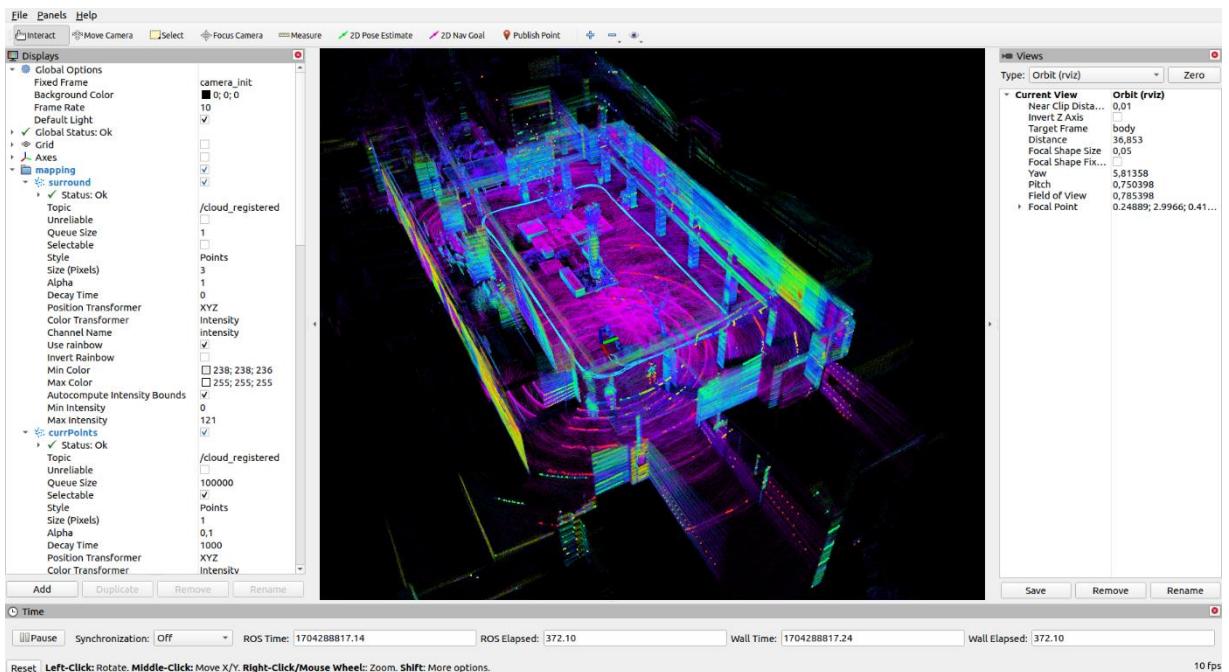


Hình 4.8. Cơ chế hoạt động Public and Subscribe của ROS

Hình 4.9 trình bày sơ đồ cấu trúc phần mềm của hệ thống thực nghiệm. Sơ đồ thể hiện các node và topic tương ứng mà các node này publish hay subscribe. Lớp Dirver sẽ phụ trách nhận dữ liệu thô từ cảm biến và chuyển đổi chúng thành những kiểu dữ liệu chuẩn, sau đó publish chúng cho những node khác. Lớp Process thực hiện tính toán ước lượng và publish kết quả cho lớp Display để hiển thị. Bên cạnh đó khi lớp Diver nhận được dữ liệu từ cảm biến, ta có thể sử dụng công cụ rosbag của ROS để lưu dữ liệu vào file bag. Các thông tin có trong file bag hoàn toàn tương đồng về mặt thời gian và dữ liệu so với các data như lúc thu về, từ đó dễ dàng phát đi phát lại thuận lợi cho việc phát triển và đánh giá thuật toán. Ngoài ra, lớp Process cũng có thể được chạy độc lập trên laptop cá nhân với dữ liệu input được cung cấp từ file bag, giúp thuận lợi trong quá trình nghiên cứu thuật toán. Về mặt hiển thị, ROS cung cấp môi trường ảo Rviz giúp minh họa tọa độ vật thể cũng như quỹ đạo của nó trong môi trường 3D, đặc biệt tái hiện lại bản độ và cấu trúc môi trường xung quanh trong không gian ba chiều rất trực quan cho người sử dụng (*Hình 4.10*).



Hình 4.9. Cấu trúc phần mềm



Hình 4.10. Giao diện Rviz

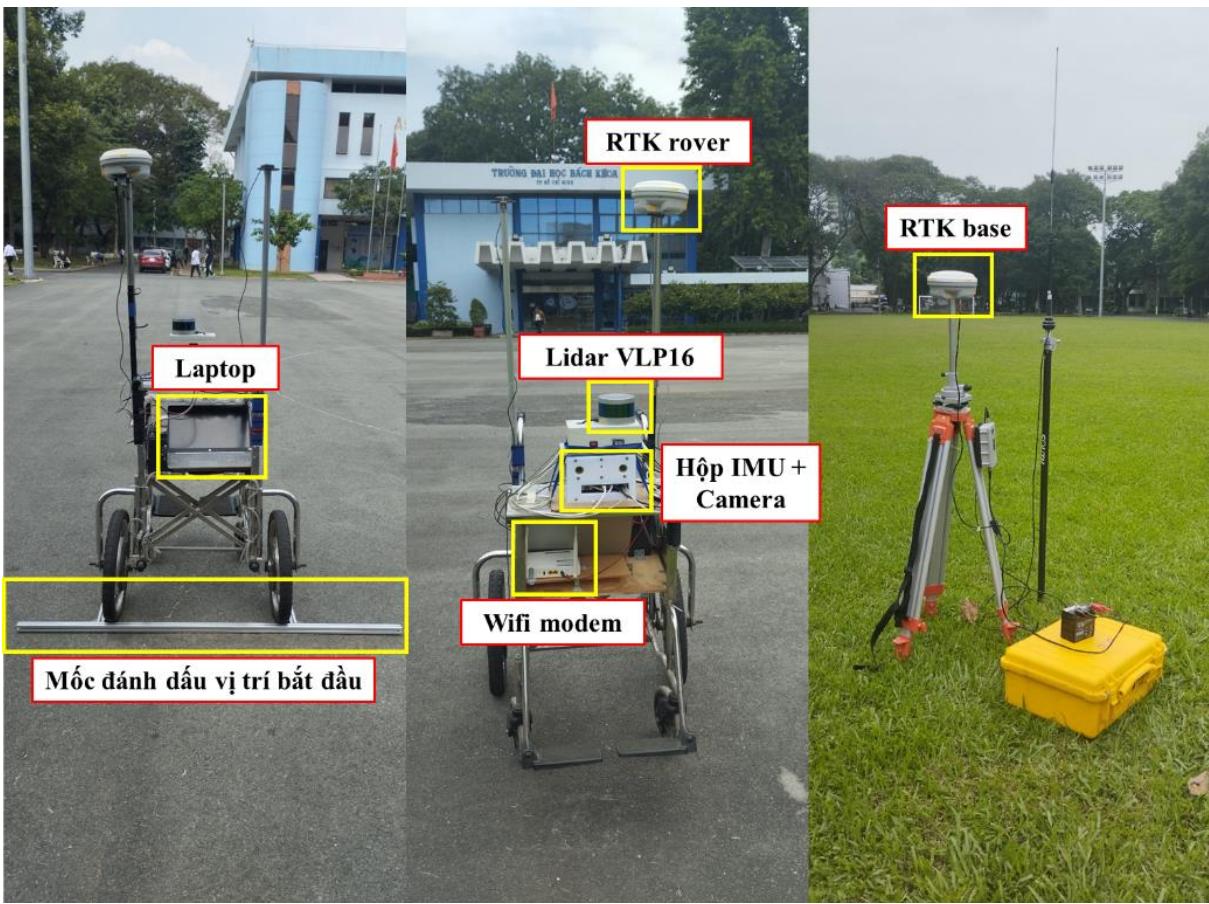
Chương 5. KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

5.1. Tổng quan

Trong đề tài này, ngoài thực nghiệm với hệ định vị tích hợp IMU/Lidar, kết quả ước lượng vị trí từ hệ định vị tích hợp IMU/Camera [54] cũng sẽ được thu thập để so sánh với kết quả với hệ IMU/Lidar. Đầu tiên các thiết bị sẽ được bố trí như *Hình 5.1* tại trường Đại học Bách Khoa TP.HCM (tên viết tắt: HCMUT). Mốc đánh dấu vị trí bắt đầu để giúp quỹ đạo di chuyển có vị trí bắt đầu và vị trí kết thúc trùng nhau, từ đó tính được sai số đầu cuối. Trạm base của RTK sẽ được bố trí cố định trên sân cỏ. Trong phần này sẽ có tổng cộng 6 tập dữ liệu được thu thập bởi hệ thống này, bao gồm hai tập dữ liệu indoor thu thập bên trong tòa nhà A4 với tốc độ di chuyển là 0.5 m và 1 m, hai tập dữ liệu outdoor phía trước tòa nhà A2 với quỹ đạo hình tròn, tốc độ di chuyển là 0.5 m và 1 m và hai tập dữ liệu outdoor phía trước tòa nhà A2 với quỹ đạo hình chữ nhật, tốc độ di chuyển là 0.5 m và 1 m. Bên cạnh đó một tập dữ liệu đi xung quanh khuôn viên trường cũng sẽ được thu thập. Các thông số được sử dụng cho các tập dữ liệu thu thập tại trường Đại học Bách Khoa TP.HCM được cho trong *Bảng 5.1*.

Bảng 5.1. Bảng thông số sử dụng cho các tập dữ liệu tại HCMUT

Ma trận hiệp phương sai Q	$q_g = 0.1, q_a = 0.1,$ $q_{bg} = 0.0001, q_{ba} = 0.0001$
Ma trận hiệp phương sai R	$R = 0.001I_{n \times n}$
Ma trận biến đổi thuần nhất từ L-frame sang I-frame ${}^I T_L$	${}^I R_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ${}^I p_L = [-0.01 \quad 0.00 \quad 0.1448]^T$



Hình 5.1. Bố trí hệ thống thực nghiệm

Ngoài ra tập dữ liệu có chứa Encoder được cung cấp từ [52] cũng sẽ được sử dụng để đánh giá thuật toán tích hợp IMU/Lidar/Encoder. Mô tả về tập dữ liệu này sẽ được trình bày trong phần sau.

5.2. Phương pháp đánh giá sai số

Về phương pháp đánh giá sai số, dữ liệu thu được từ hệ thống định vị sẽ được căn chỉnh về hệ tọa độ GPS dùng phương pháp bình phương cực tiểu. Tiếp theo đó chênh lệch giữa quỹ đạo GPS và quỹ đạo có được từ bộ ước lượng sẽ được tính bằng sai số hiệu dụng RMS.

$$e_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{est}[i] - x_{ref}[i])^2} \quad (5.1)$$

Bên cạnh sai số RMS, các quỹ đạo có vị trí bắt đầu và vị trí kết thúc trùng nhau sẽ được đánh giá thêm bởi tiêu chí sai số đầu cuối (công thức (5.2)). Một tiêu chí đánh giá quan trọng nữa cũng sẽ được sử dụng trong phần này, đó là phần trăm sai số trên tổng quãng đường đi được tại thời điểm kết thúc (công thức (5.3)).

$$e_{end to end} = \|x_{end} - x_{start}\| \quad (5.2)$$

$$e_{err \ per \ distance}[end] = \frac{\|x_{est}[end] - x_{ref}[end]\|}{distance[end]} * 100\% \quad (5.3)$$

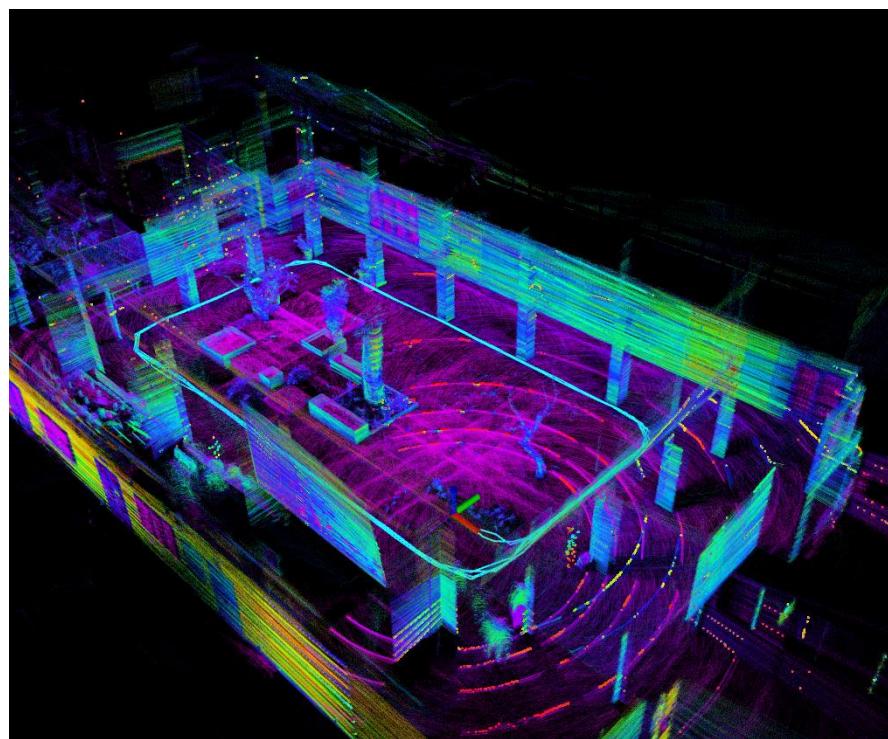
Ngoài ra đối với các tập dữ liệu được thu thập tại trường Đại học Bách Khoa TP.HCM, các góc hướng sẽ được tham chiếu với giá trị góc trả về từ IMU nhưng chỉ mang tính chất tham khảo do độ chính xác dữ liệu góc từ IMU không cao, chỉ nhằm mục đích cho thấy các góc này không bị trôi trong quá trình ước lượng.

5.3. Kết quả thực nghiệm indoor

Tập dữ liệu được thu thập bên trong tòa nhà A4.



Hình 5.2. Bên trong tòa nhà A4



Hình 5.3. Bản đồ dựng được bên trong tòa nhà A4

5.3.1. Tập dữ liệu di chuyển nhanh

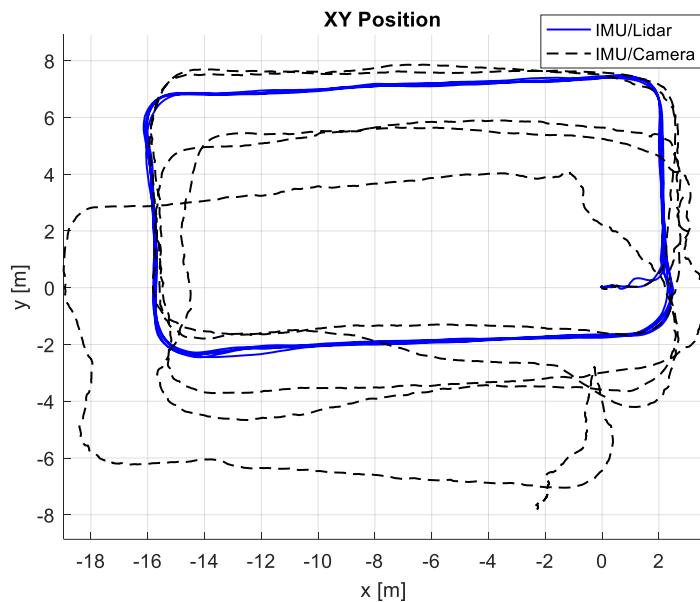
Bảng 5.2. Thông tin tập dữ liệu indoor di chuyển nhanh

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
272.971 m	248.415 s	1.099 m/s

Bảng 5.3. Kết quả thực nghiệm tập dữ liệu indoor di chuyển nhanh

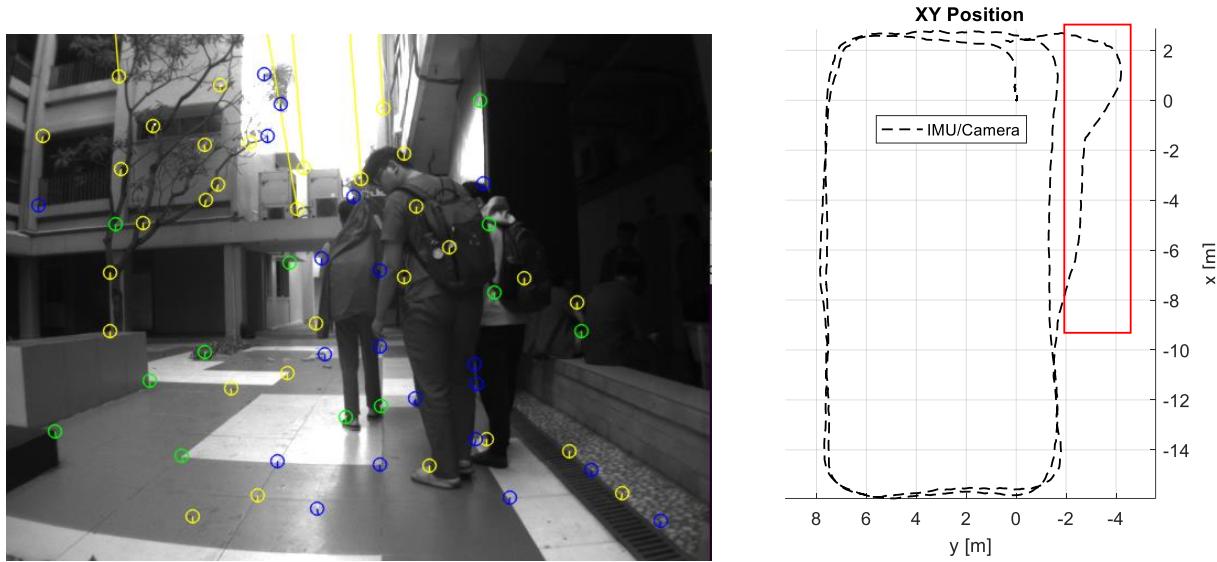
	IMU/Lidar	IMU/Camera
Sai số đầu cuối về khoảng cách	0.035 m	8.156 m
Sai số đầu cuối về chiều cao	0.015 m	1.812 m
Sai số đầu cuối về góc yaw	0.169 độ	19.467 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.012 %	2.933 %

Có thể thấy quỹ đạo của hệ IMU/Lidar tốt hơn nhiều so với hệ IMU/Camera, bên cạnh đó các trạng thái của hệ IMU/Lidar hầu như quay trở lại như cũ gần như trùng với vị trí ban đầu, các trạng thái của hệ IMU/Camera thì bị trôi đáng kể.

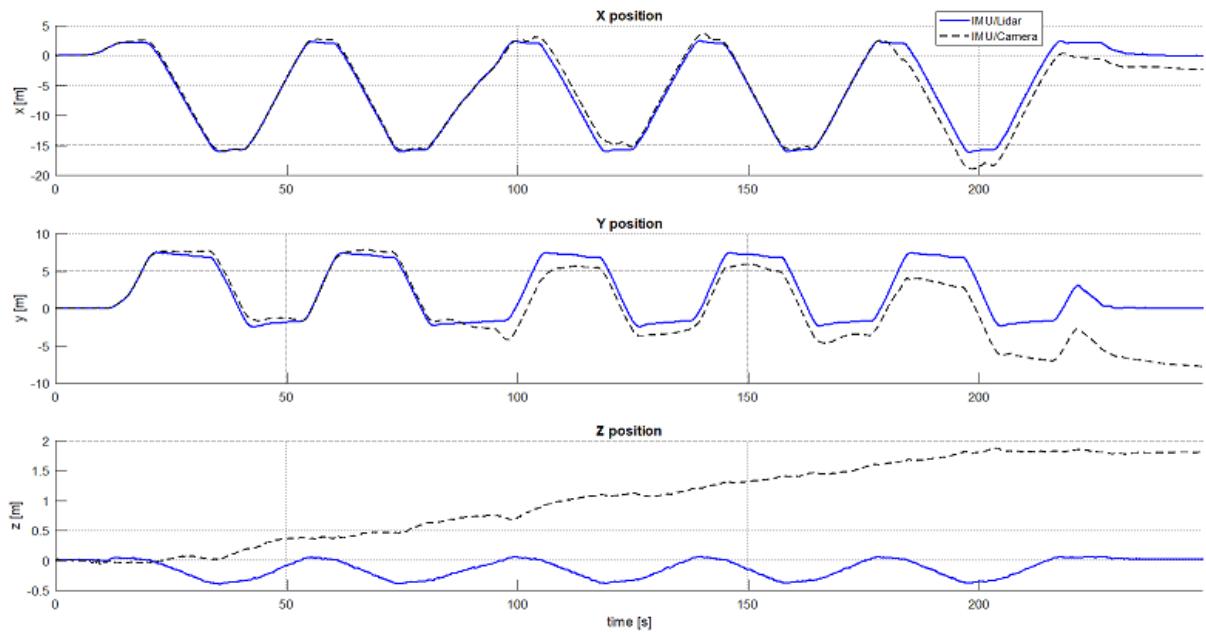


Hình 5.4. Quỹ đạo 2D tập dữ liệu indoor di chuyển nhanh

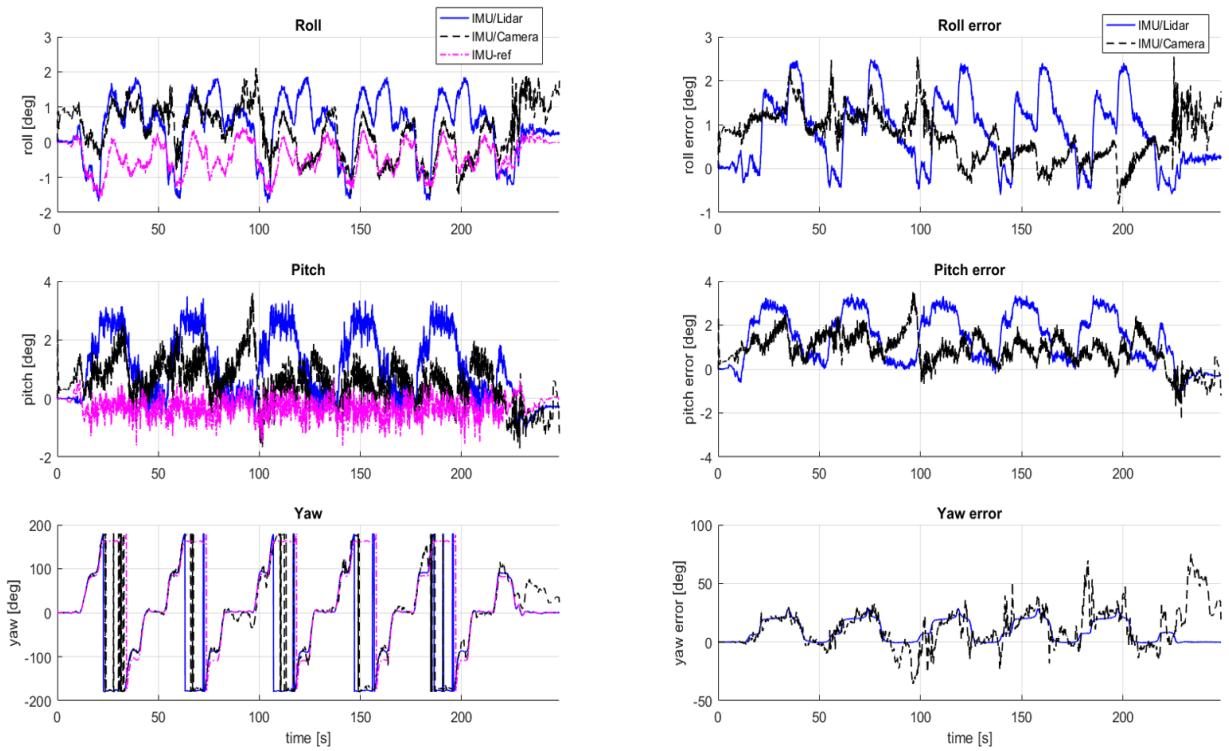
Ngoài ra trong quá trình thu thập dữ liệu quỹ đạo của hệ IMU/Camera còn bị lệch do có một nhóm sinh viên di chuyển qua (phản ứng khung trong viền đỏ *Hình 5.5*) trong khi quỹ đạo của hệ IMU/Lidar vẫn duy trì được độ chính xác.



Hình 5.5. Quỹ đạo IMU/Camera bị lệch do nhóm sinh viên di chuyển



Hình 5.6. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển nhanh



Hình 5.7. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển nhanh

5.3.2. Tập dữ liệu di chuyển chậm

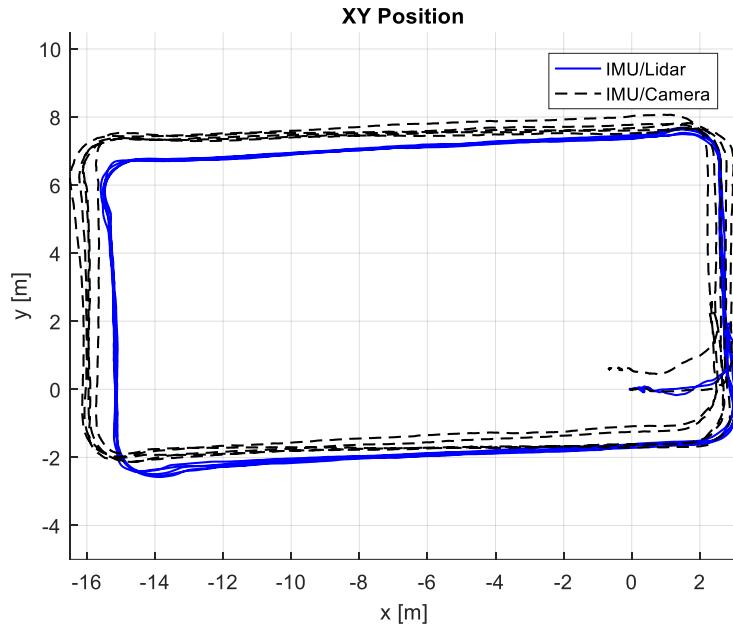
Bảng 5.4. Thông tin tập dữ liệu indoor di chuyển nhanh

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
276.835 m	459.916 s	0.601 m/s

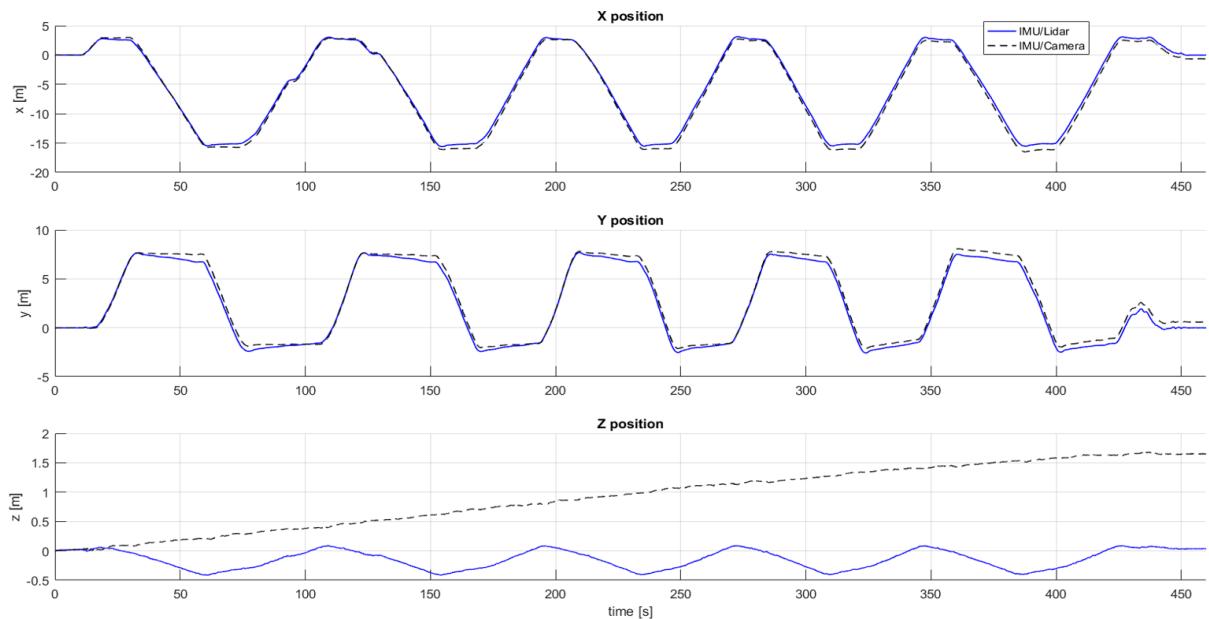
Bảng 5.5. Kết quả thực nghiệm tập dữ liệu indoor di chuyển nhanh

	IMU/Lidar	IMU/Camera
Sai số đầu cuối về khoảng cách	0.027 m	0.891 m
Sai số đầu cuối về chiều cao	0.036 m	1.648 m
Sai số đầu cuối về góc yaw	0.167 độ	2.793 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.010 %	0.313 %

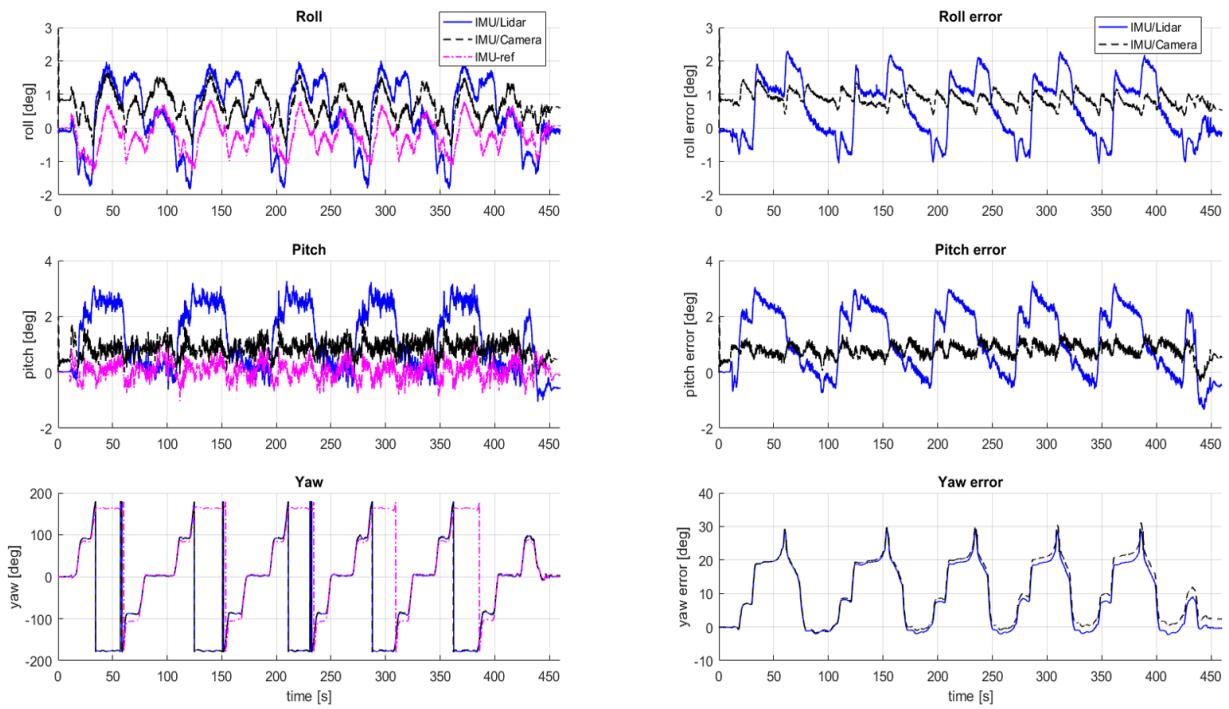
Khi di chuyển chậm, mặc dù trạng thái của hệ IMU/Camera không bị trôi nhiều nhưng quỹ đạo của hệ IMU/Lidar tốt hơn so với hệ IMU/Camera.



Hình 5.8. Quỹ đạo 2D tập dữ liệu indoor di chuyển chậm



Hình 5.9. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển chậm



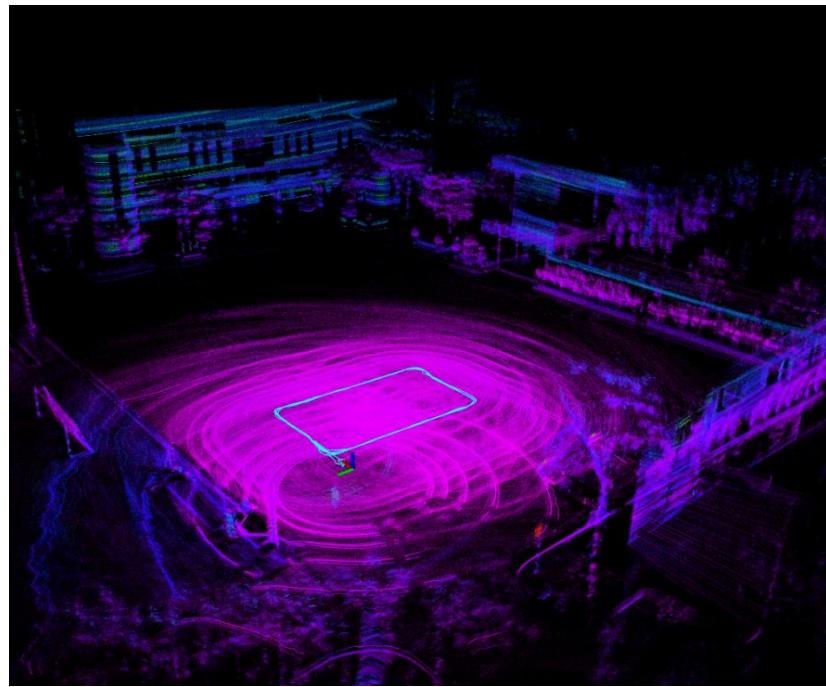
Hình 5.10. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu indoor di chuyển chậm

5.4. Kết quả thực nghiệm outdoor

Thí nghiệm outdoor được thực hiện phía trước tòa nhà A2, đoạn đường lúc này sẽ gò ghè hơn so với bên trong tòa nhà A4.



Hình 5.11. Phía trước tòa nhà A2



Hình 5.12. Bản đồ dựng được phía trước tòa nhà A2

5.4.1. Tập dữ liệu di chuyển theo hình tròn

5.4.1.1. Di chuyển nhanh

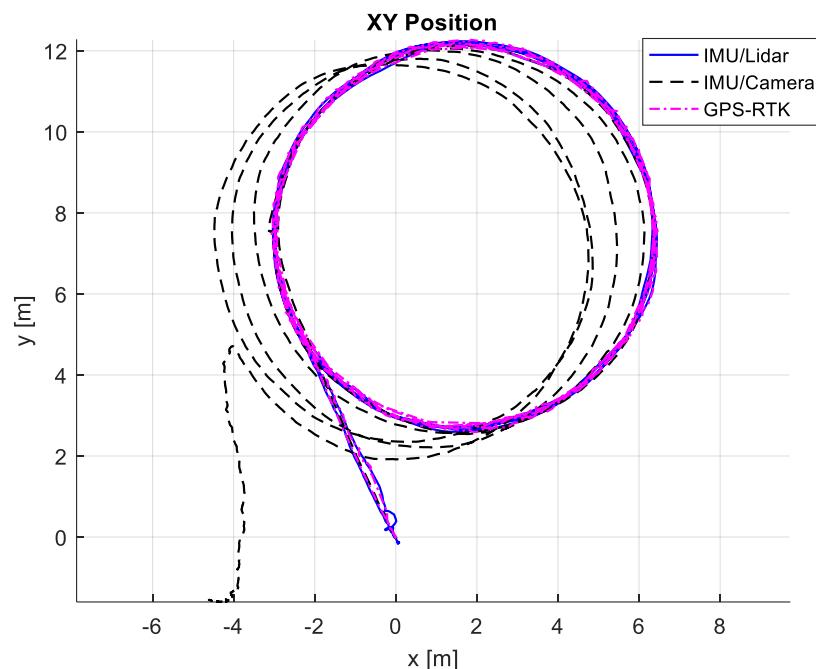
Bảng 5.6. Thông tin tập dữ liệu outdoor, hình tròn, di chuyển nhanh

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
166.305 m	149.372 s	1.113 m/s

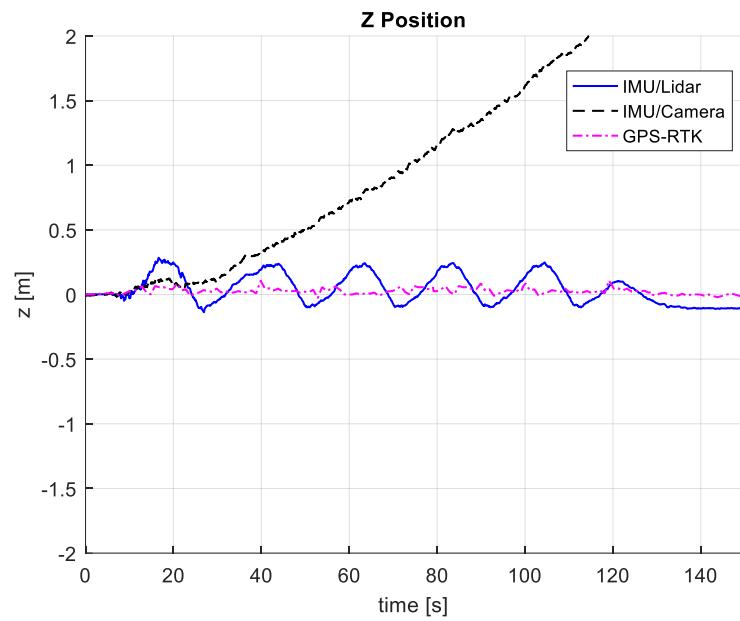
Bảng 5.7. Kết quả thực nghiệm tập dữ liệu outdoor, hình tròn, di chuyển nhanh

	IMU/Lidar	IMU/Camera
Sai số RMS chênh lệch quỹ đạo	0.085 m	1.994 m
Sai số đầu cuối về khoảng cách	0.138 m	4.868 m
Sai số đầu cuối về chiều cao	0.109 m	2.036 m
Sai số đầu cuối về góc yaw	0.852 độ	11.886 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.074 %	3.028 %

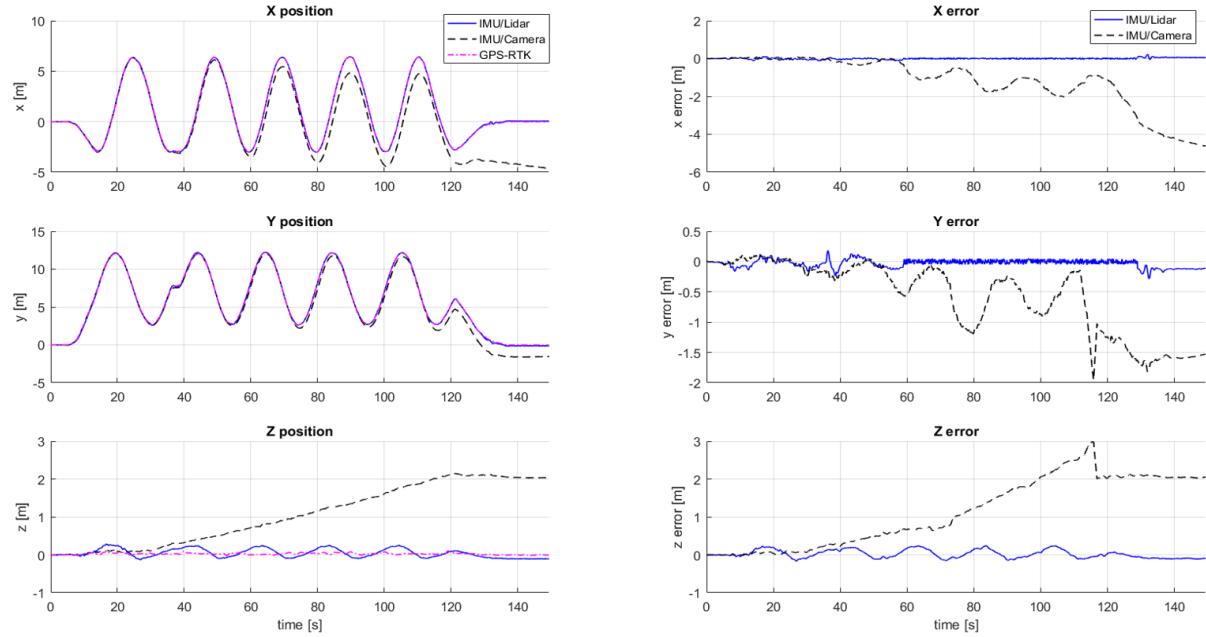
Có thể thấy khi di chuyển ngoài trời với tốc độ nhanh, di chuyển với quỹ đạo phức tạp như quỹ đạo hình tròn do phải chuyển hướng liên tục, hệ IMU/Lidar vẫn giữ được độ chính xác tốt, quỹ đạo hệ IMU/Camera thì ngày càng lệch. Ngoài ra khi quan sát dữ liệu vị trí trên trục z, mặc dù hệ IMU/Lidar xuất hiện sự dao động nhưng vẫn có thể quay lại giá trị tại thời điểm ban đầu, trong khi đó sai số trục z của hệ IMU/Camera lại tăng theo thời gian.



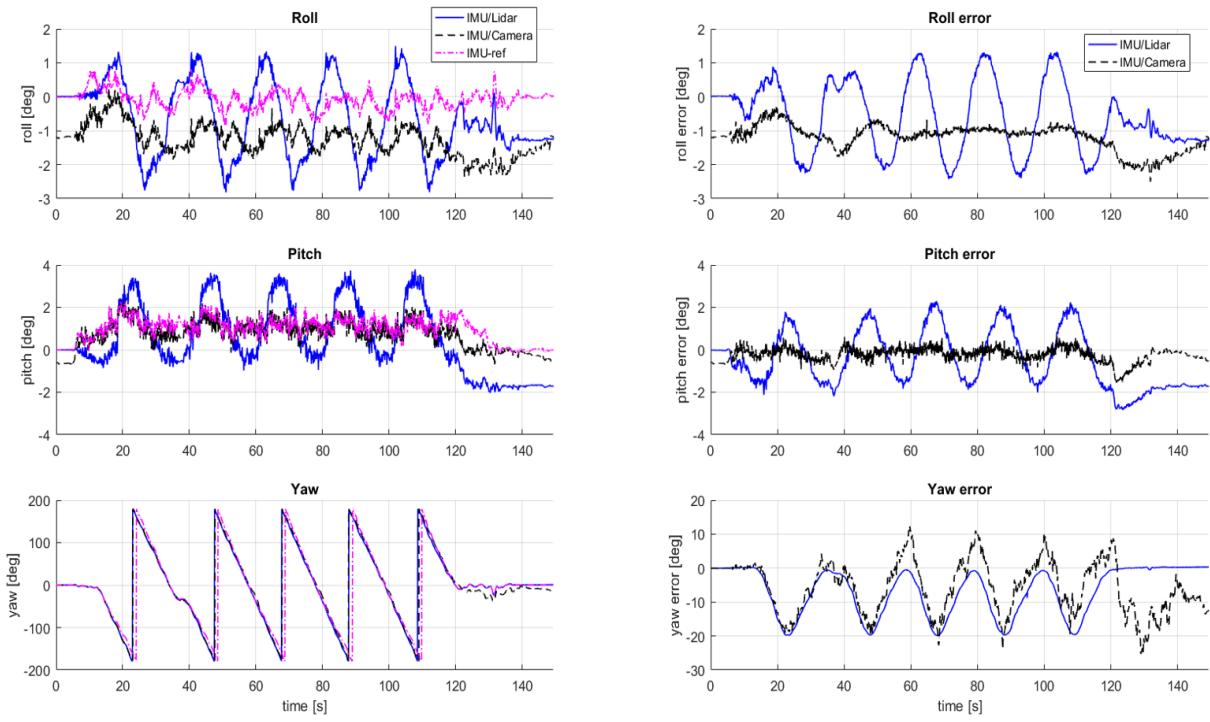
Hình 5.13. Quỹ đạo 2D tập dữ liệu outdoor hình tròn di chuyển nhanh



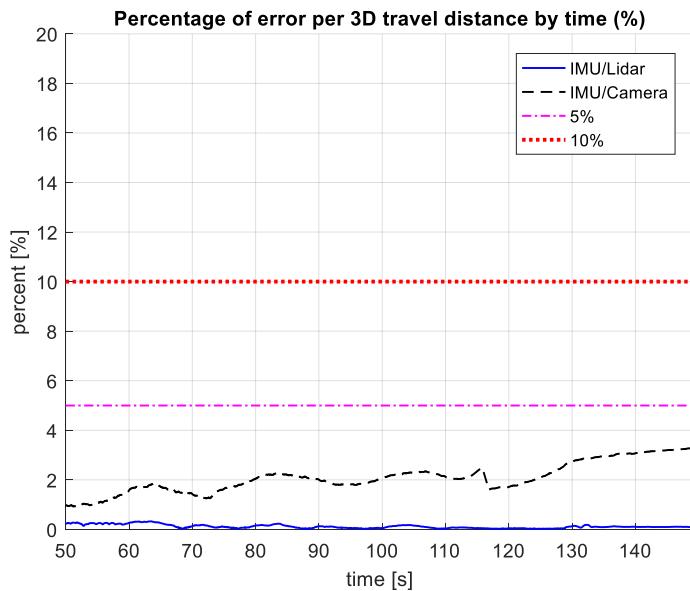
Hình 5.14. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh



Hình 5.15. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh



Hình 5.16. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh



Hình 5.17. Phần trăm sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển nhanh

5.4.1.2. Di chuyển chậm

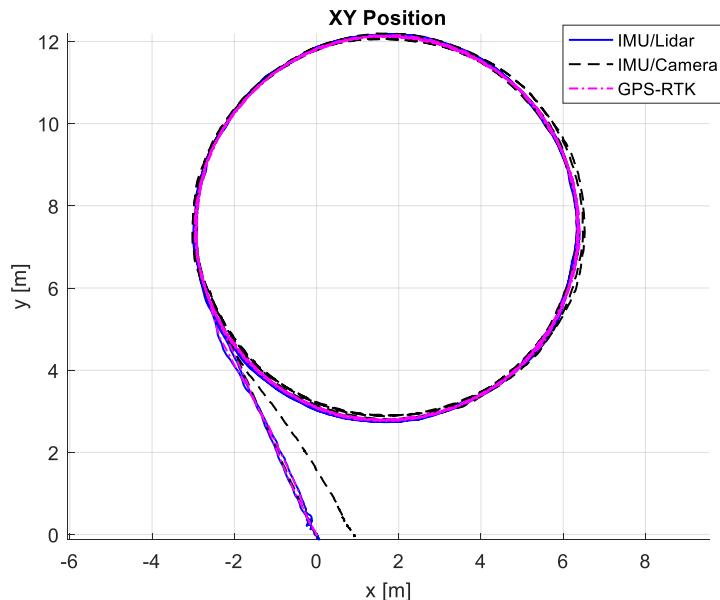
Bảng 5.8. Thông tin tập dữ liệu outdoor, hình tròn, di chuyển chậm

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
161.312 m	333.540 s	0.484 m/s

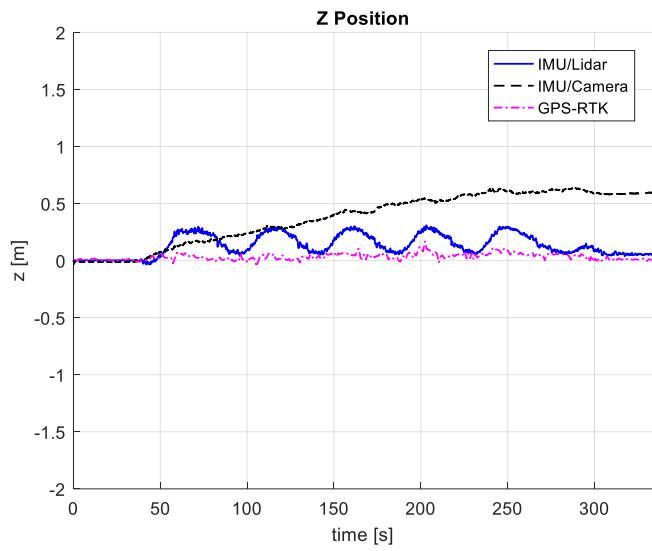
Bảng 5.9. Kết quả thực nghiệm tập dữ liệu outdoor, hình tròn, di chuyển chậm

	IMU/Lidar	IMU/Camera
Sai số RMS chênh lệch quỹ đạo	0.058 m	0.389 m
Sai số đầu cuối về khoảng cách	0.098 m	0.914 m
Sai số đầu cuối về chiều cao	0.061 m	0.593 m
Sai số đầu cuối về góc yaw	1.404 độ	5.110 độ
Phản trắc sai số theo quãng đường tại thời điểm kết thúc	0.060 %	0.568 %

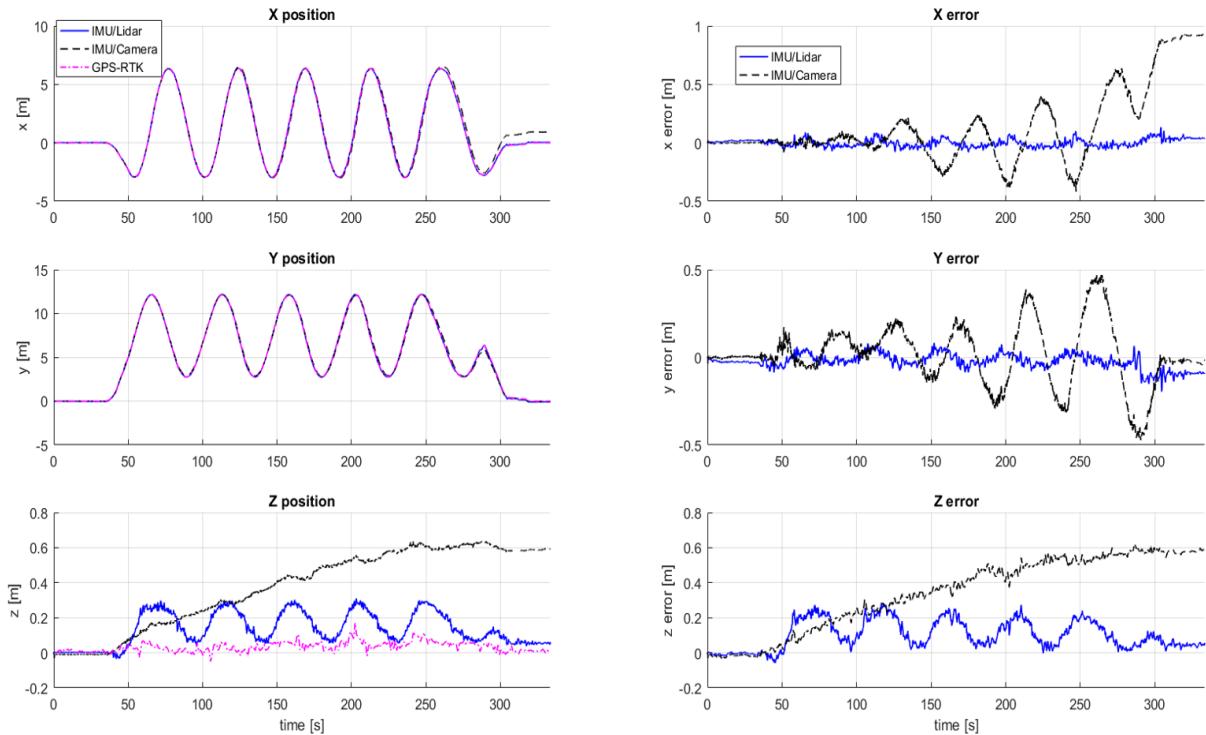
Khi di chuyển với tốc độ chậm, hệ IMU/Camera ít bị trôi hơn, dữ liệu vị trí trực z tốt hơn nhưng hệ IMU/Lidar vẫn có độ chính xác cao hơn.



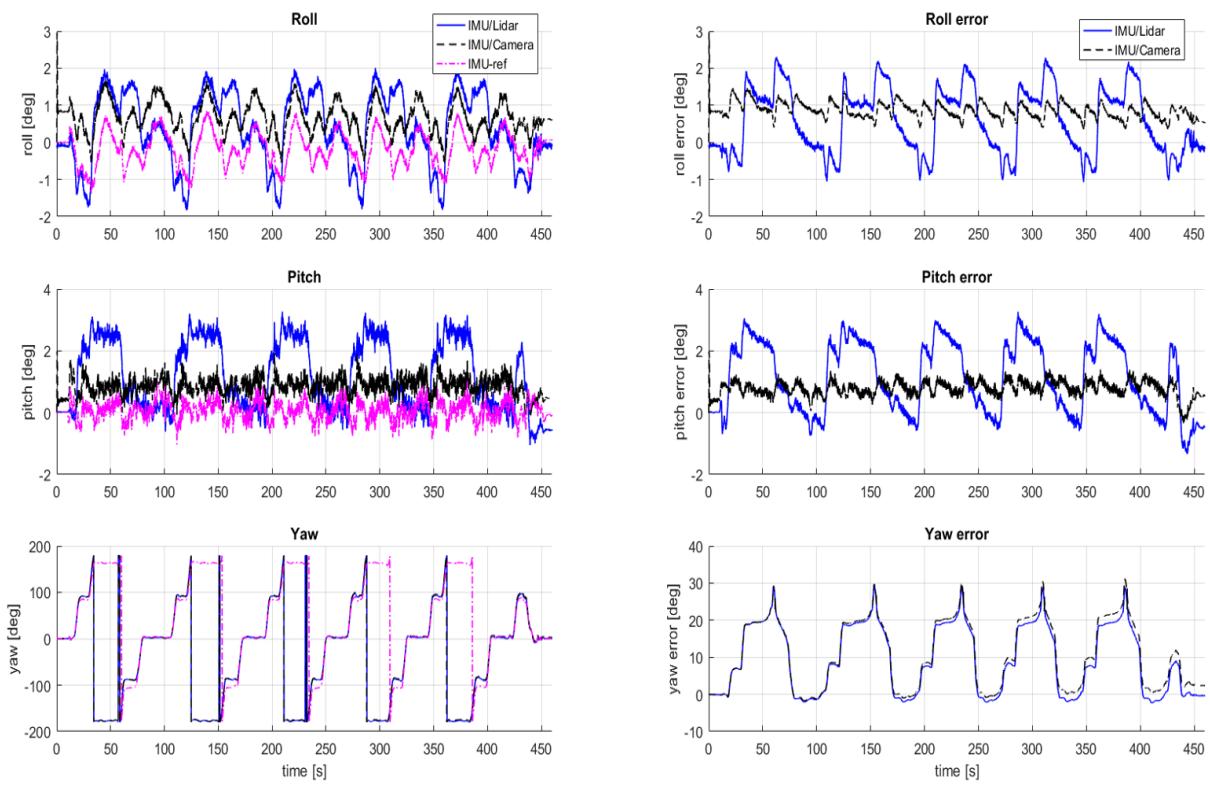
Hình 5.18. Quỹ đạo 2D tập dữ liệu outdoor hình tròn di chuyển chậm



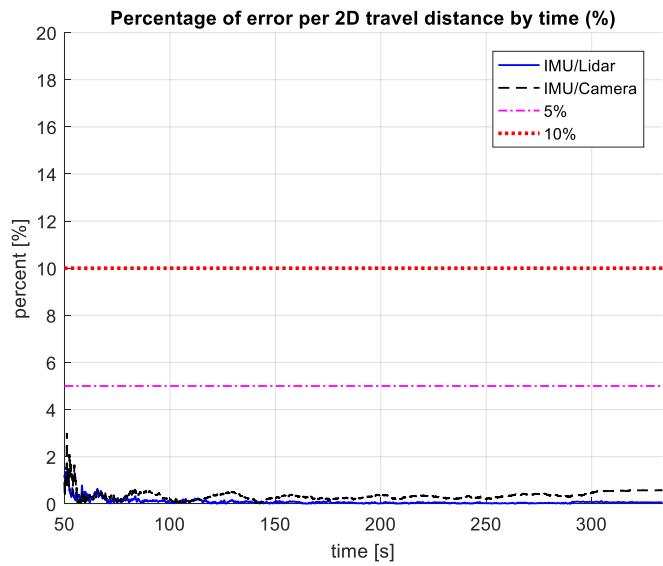
Hình 5.19. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm



Hình 5.20. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm



Hình 5.21. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm



Hình 5.22. Phản trăng sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình tròn di chuyển chậm

5.4.2. Tập dữ liệu di chuyển theo hình chữ nhật

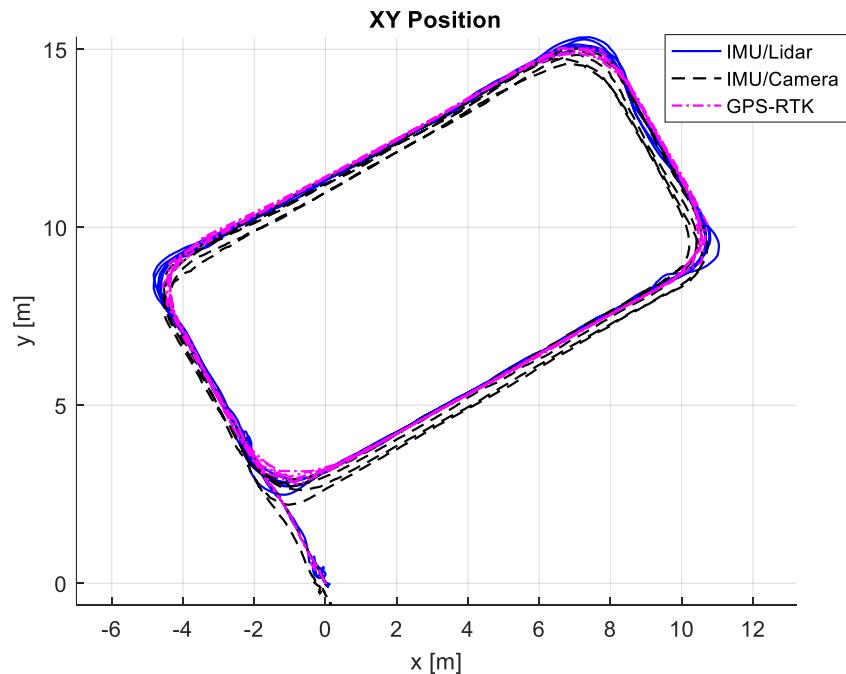
5.4.2.1. Di chuyển nhanh

Bảng 5.10. Thông tin tập dữ liệu outdoor, hình chữ nhật, di chuyển nhanh

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
212.546 m	215.842 s	0.985 m/s

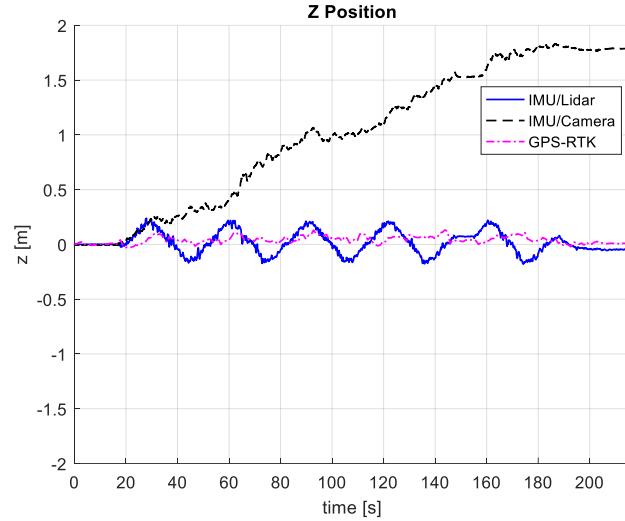
Bảng 5.11. Kết quả thực nghiệm tập dữ liệu outdoor, hình chữ nhật, di chuyển nhanh

	IMU/Lidar	IMU/Camera
Sai số RMS chênh lệch quỹ đạo	0.123 m	0.360 m
Sai số đầu cuối về khoảng cách	0.071 m	0.596 m
Sai số đầu cuối về chiều cao	0.043 m	1.788 m
Sai số đầu cuối về góc yaw	0.017 độ	0.819 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.032 %	0.281 %

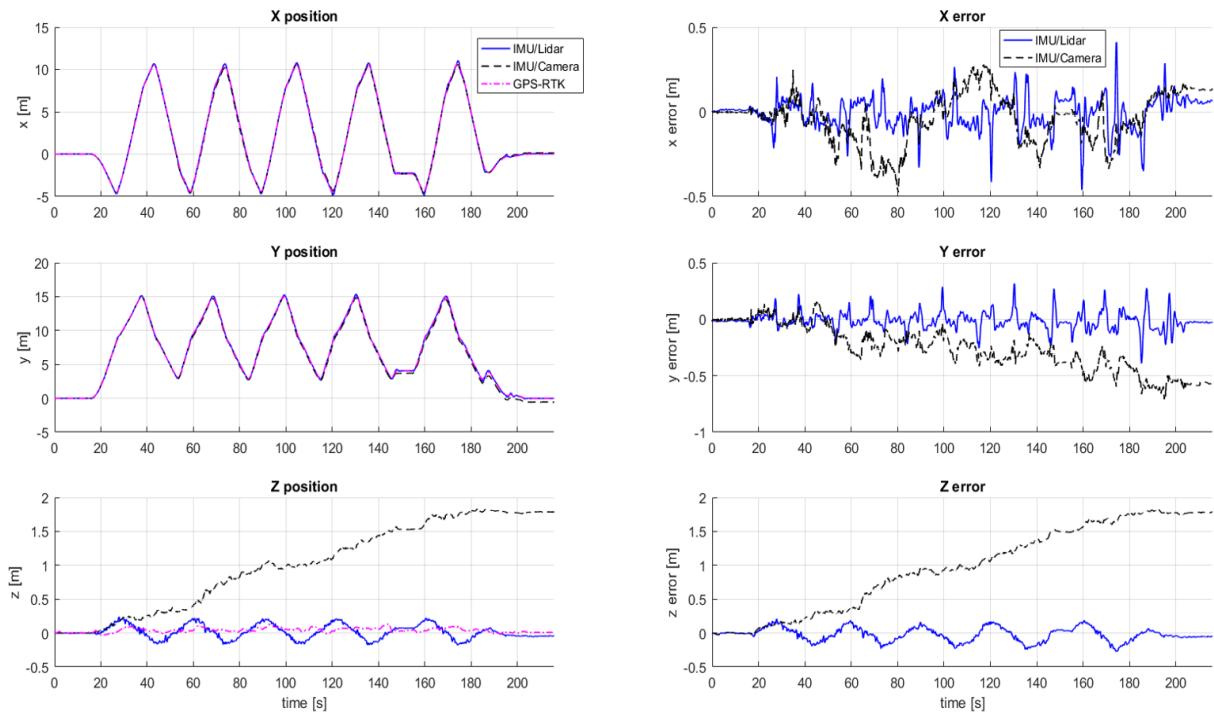


Hình 5.23. Quỹ đạo 2D tập dữ liệu outdoor hình chữ nhật di chuyển nhanh

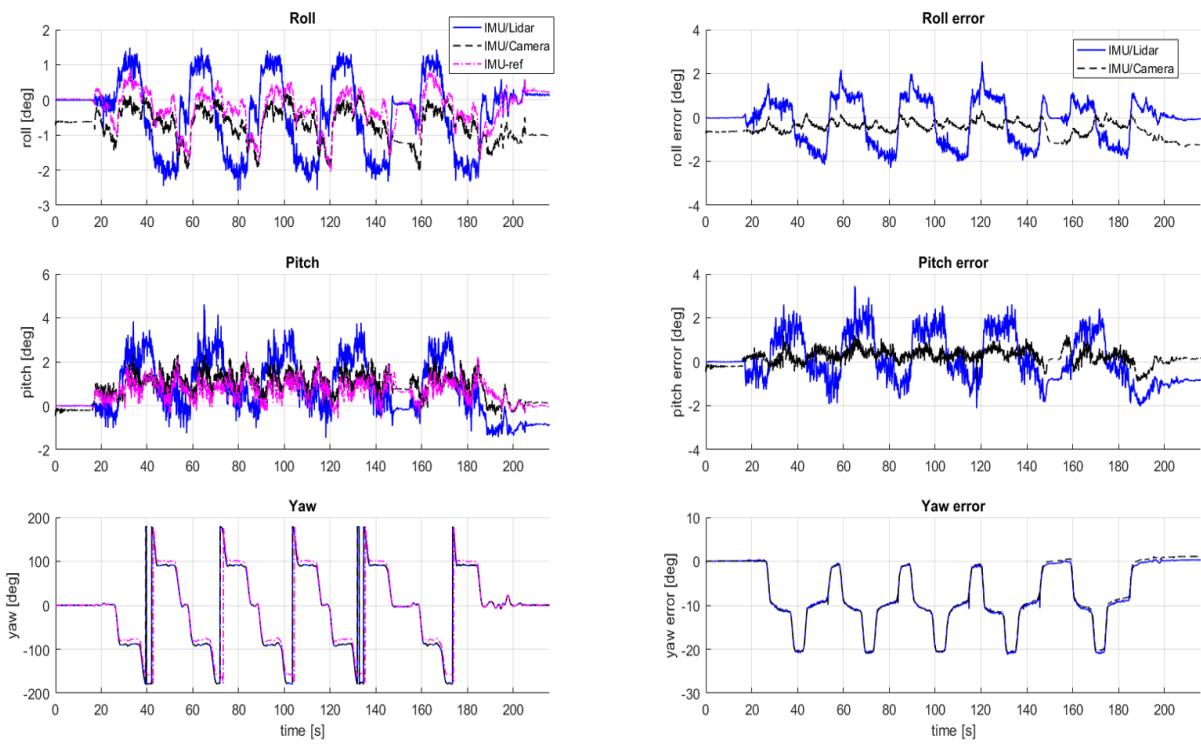
Với quỹ đạo hình chữ nhật, độ chính xác của hệ IMU/Camera tăng lên đáng kể do không phải thay đổi góc liên tục như trong quỹ đạo hình tròn nhưng vị trí trực z vẫn trôi.



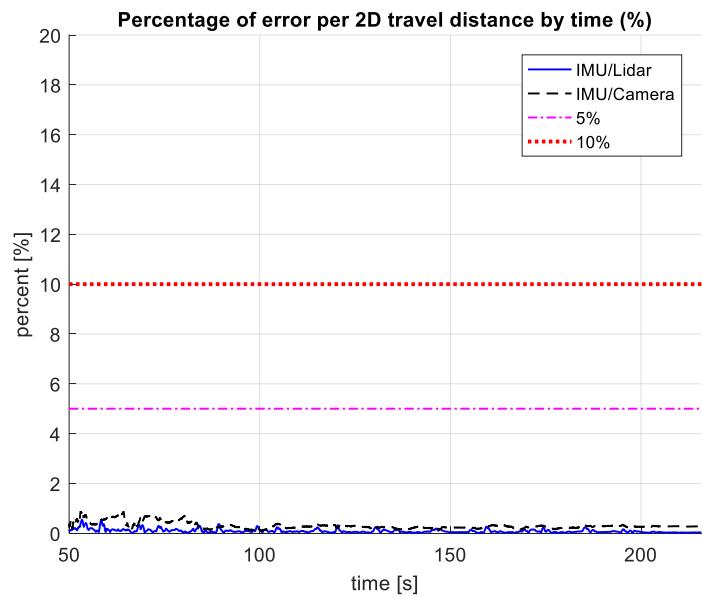
Hình 5.24. Dữ liệu vị trí trực z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh



Hình 5.25. So sánh dữ liệu vị trí trực x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh



Hình 5.26. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh



Hình 5.27. Phản trăm sai só theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển nhanh

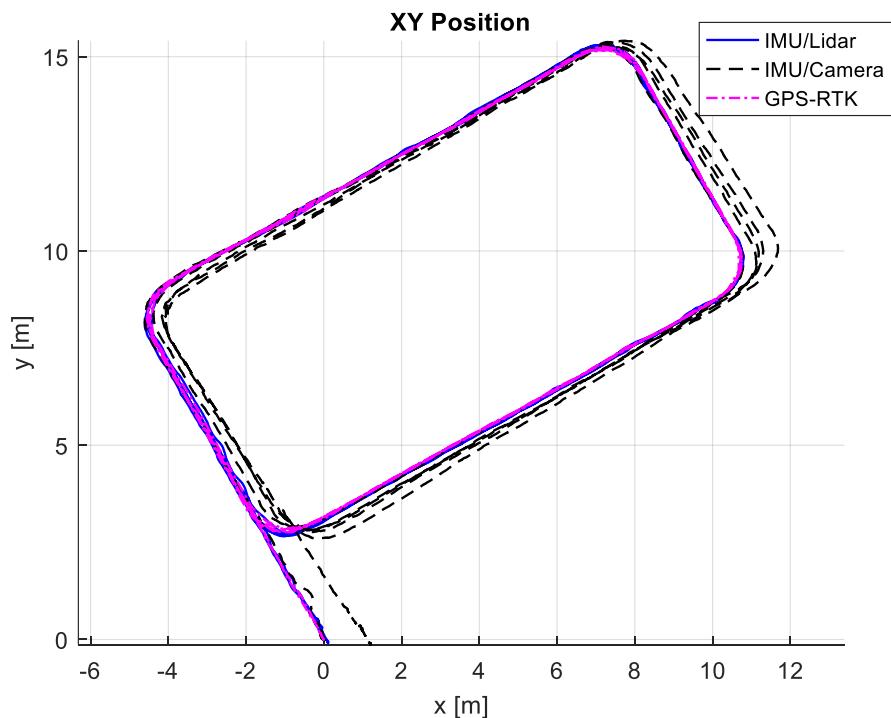
5.4.2.2. Di chuyển chậm

Bảng 5.12. Thông tin tập dữ liệu outdoor, hình chữ nhật, di chuyển chậm

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
214.269 m	402.729 s	0.532 m/s

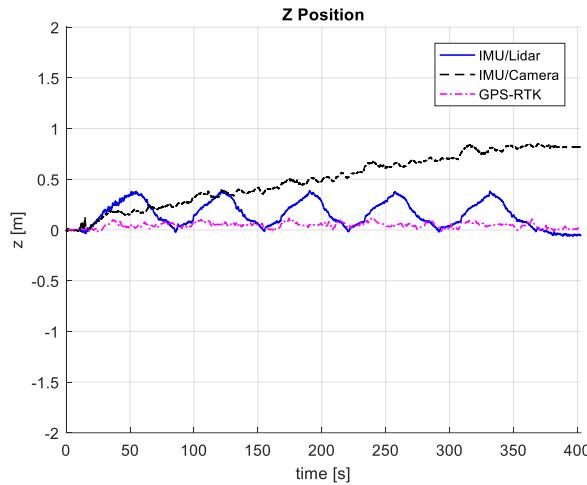
Bảng 5.13. Kết quả thực nghiệm tập dữ liệu outdoor, hình chữ nhật, di chuyển chậm

	IMU/Lidar	IMU/Camera
Sai số RMS chênh lệch quỹ đạo	0.064 m	0.600 m
Sai số đầu cuối về khoảng cách	0.102 m	1.213 m
Sai số đầu cuối về chiều cao	0.048 m	0.814 m
Sai số đầu cuối về góc yaw	0.132 độ	2.663 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.050 %	0.570 %

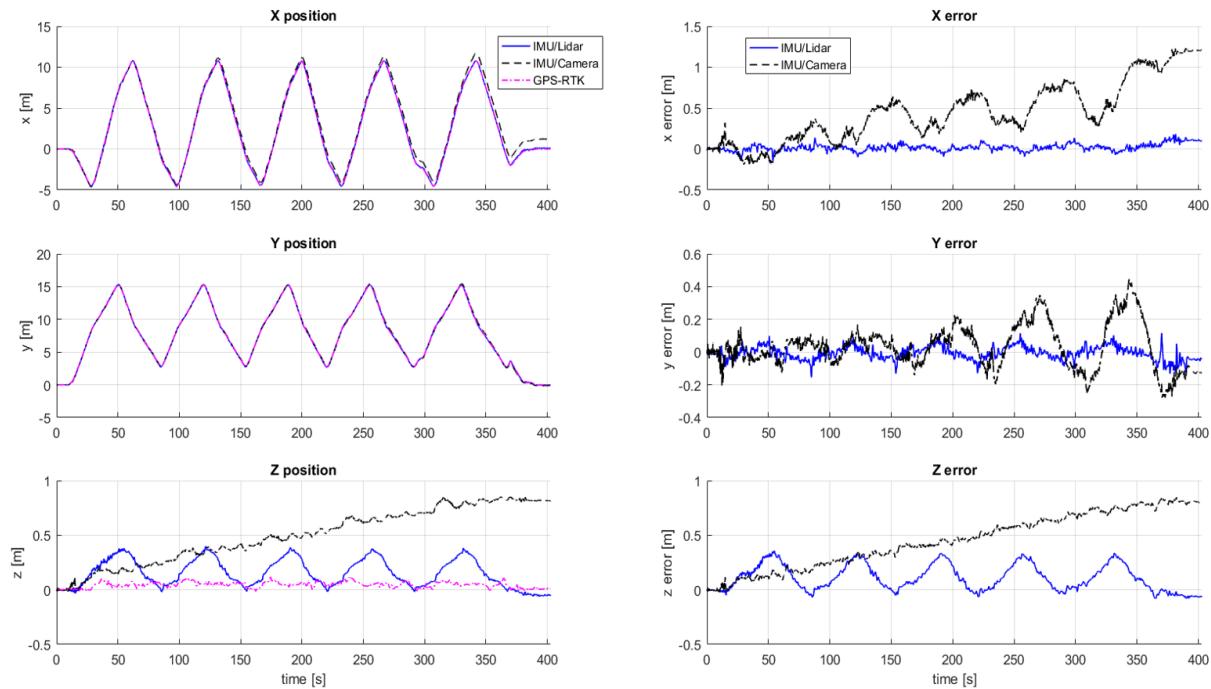


Hình 5.28. Quỹ đạo 2D tập dữ liệu outdoor hình chữ nhật di chuyển chậm

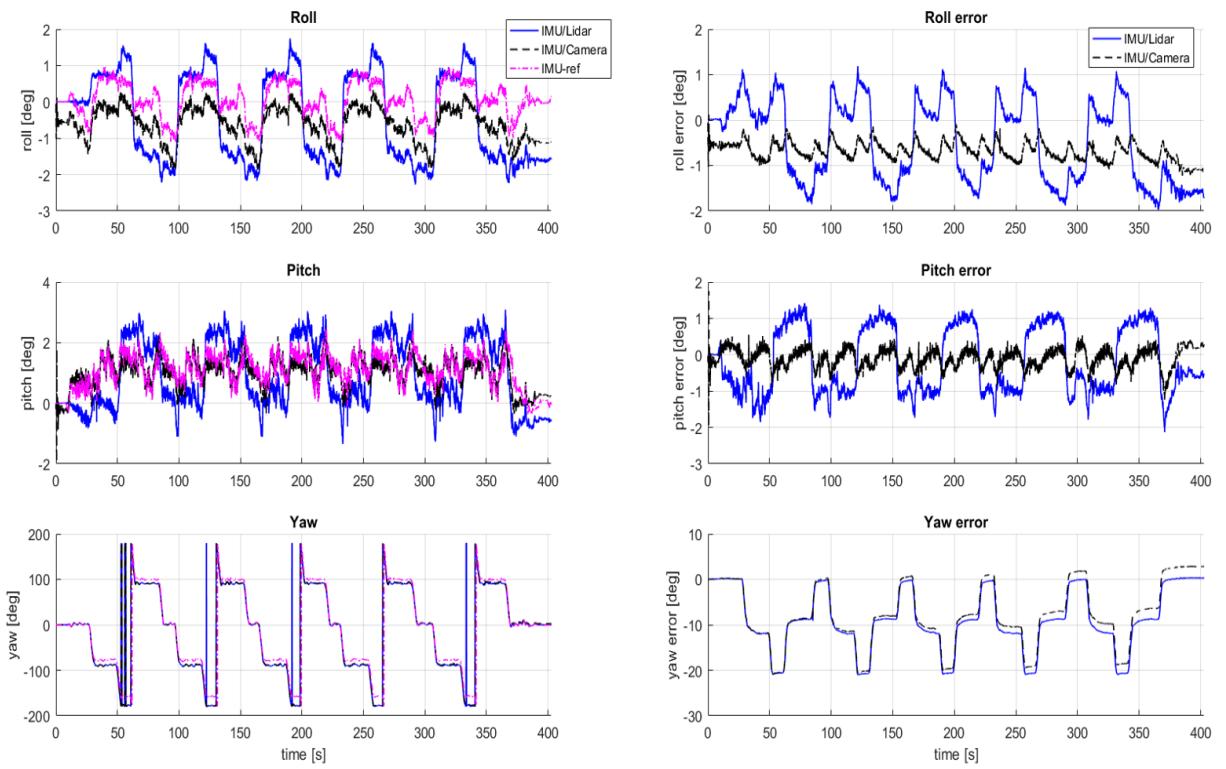
Với tập dữ liệu này hệ IMU/Camera sai số RMS của quỹ đạo có cao hơn lúc di chuyển nhanh một chút, nhưng vị trí trục z ít trôi hơn so với lúc di chuyển nhanh. Và một lần nữa, hệ IMU/Lidar vẫn cho kết quả ước lượng tốt hơn.



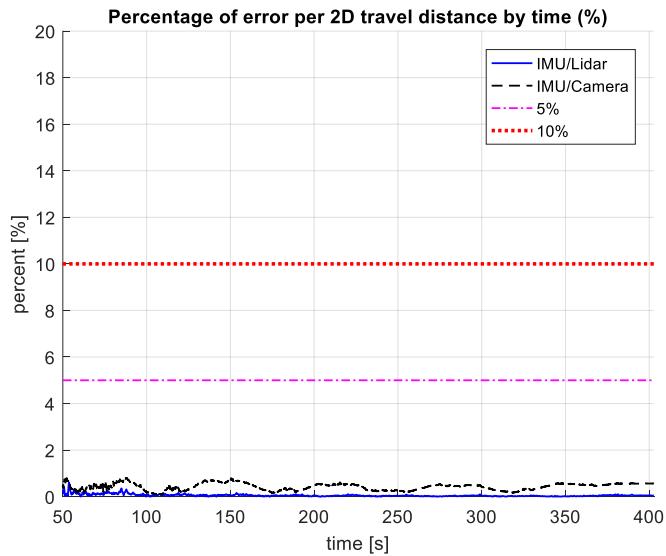
Hình 5.29. Dữ liệu vị trí trục z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm



Hình 5.30. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm



Hình 5.31. Dữ liệu góc hướng của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm



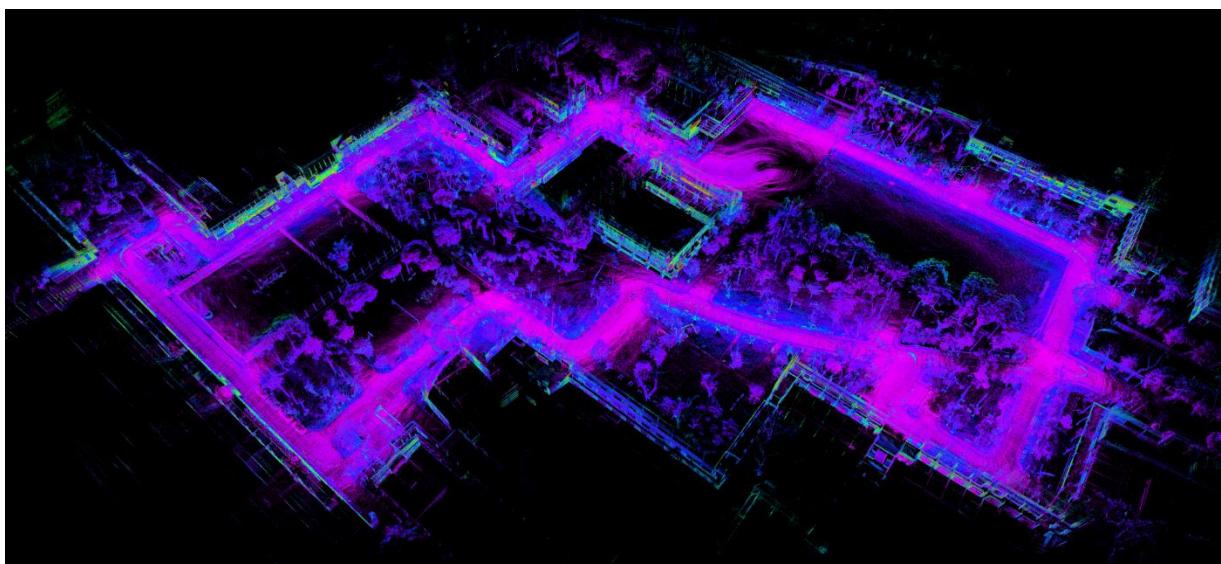
Hình 5.32. Phản trăng sai số theo quãng đường của IMU/Lidar và IMU/Camera tập dữ liệu outdoor hình chữ nhật di chuyển chậm

5.4.3. Tập dữ liệu di chuyển xung quanh khuôn viên trường HCMUT

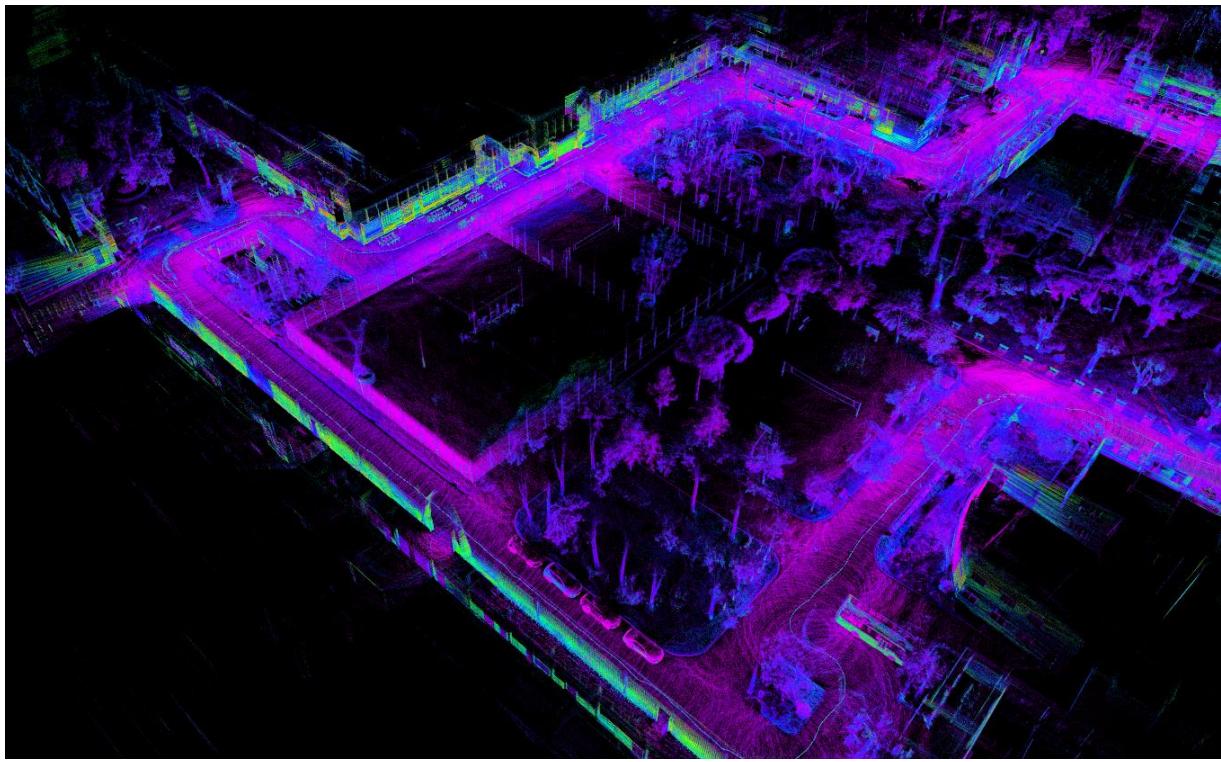
Trong tập dữ liệu này, ta sẽ di chuyển xung quanh khuôn viên trường đại học Bách Khoa. Tại thời điểm xuất phát ta sẽ di chuyển theo hình tròn 5 vòng sau đó di chuyển xung quanh khuôn viên trường, và khi trở lại khu vực xuất phát, ta sẽ sẽ di chuyển thêm 5 vòng hình tròn nữa và dừng lại ngay vị trí bắt đầu.



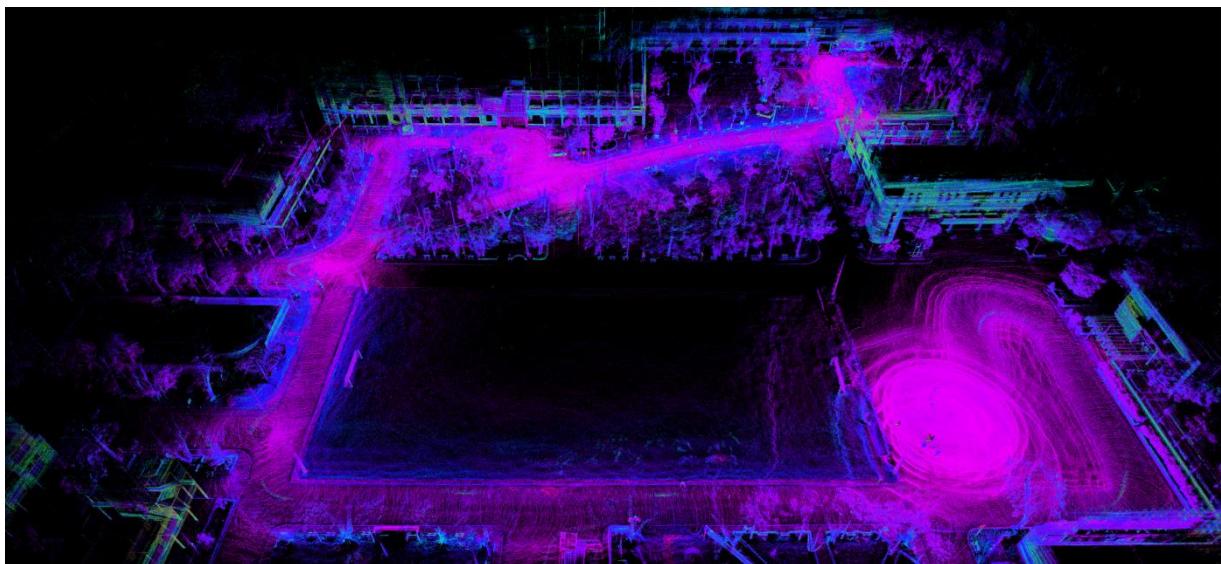
Hình 5.33. Khuôn viên trường Đại học Bách Khoa TP.HCM



Hình 5.34. Bản đồ dựng được của khuôn viên trường HCMUT



Hình 5.35. Một phần bản đồ khu vực B9, B8, B11, C3, C4, C1 trường HCMUT



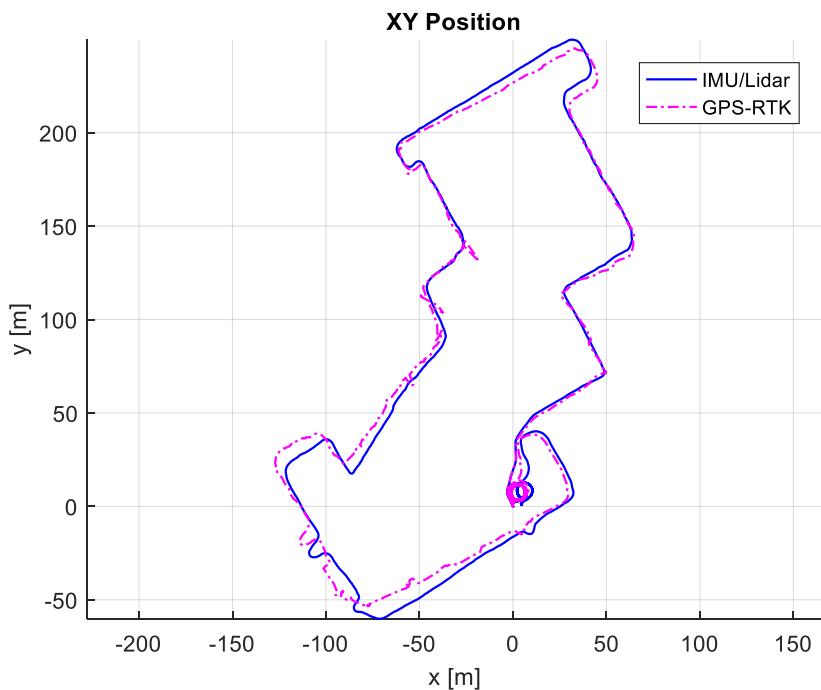
Hình 5.36. Một phần bản đồ khu vực B6, B4, B3, B2, A2, A5 trường HCMUT

Bảng 5.14. Thông tin tập dữ liệu khuôn viên trường HCMUT

Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
1330.201 m	1351.909 s	0.983 m/s

Bảng 5.15. Kết quả thực nghiệm tập dữ liệu khuôn viên trường HCMUT

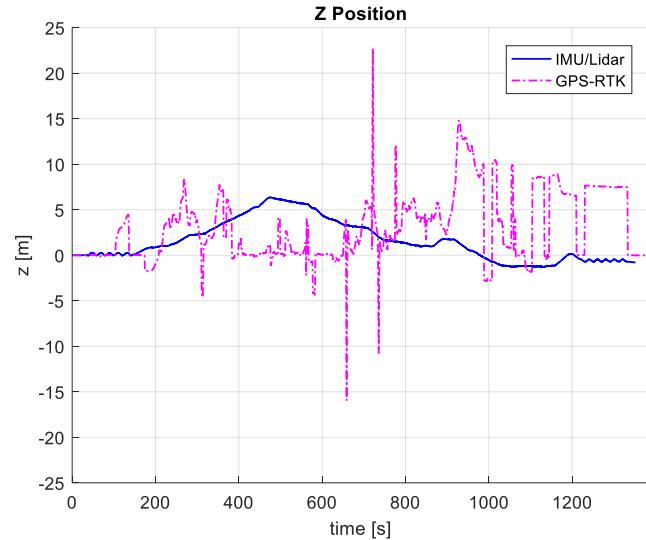
	IMU/Lidar
Sai số đầu cuối về khoảng cách	4.822 m
Sai số đầu cuối về chiều cao	0.766 m
Sai số đầu cuối về góc yaw	3.184 độ
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.362 %



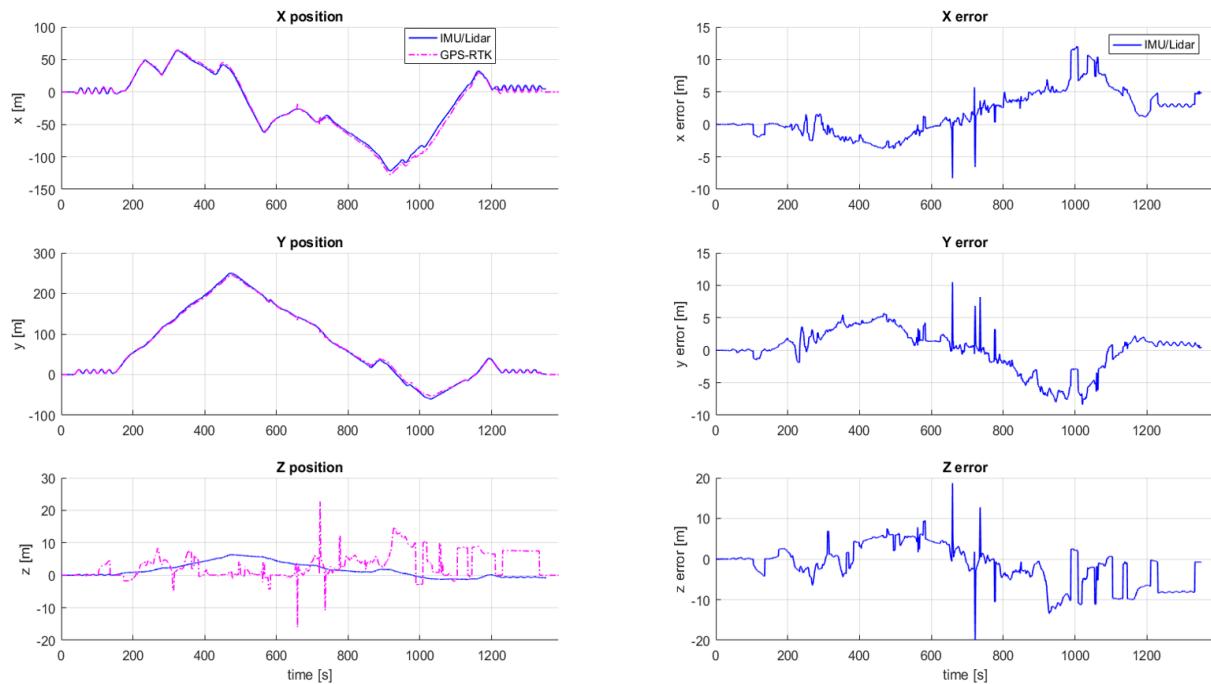
Hình 5.37. Quỹ đạo 2D tập dữ liệu outdoor khuôn viên HCMUT

Do đoạn đường đi sốc, bánh xe phía trước nhỏ dễ lọt vào các lỗ trên đường đi nên thuật toán IMU/Camera nhanh chóng bị trôi và không giữ được độ chính xác nên trong phần này ta chỉ xét đến kết quả thuật toán tích hợp IMU/Lidar. Ngoài ra do phải di chuyển

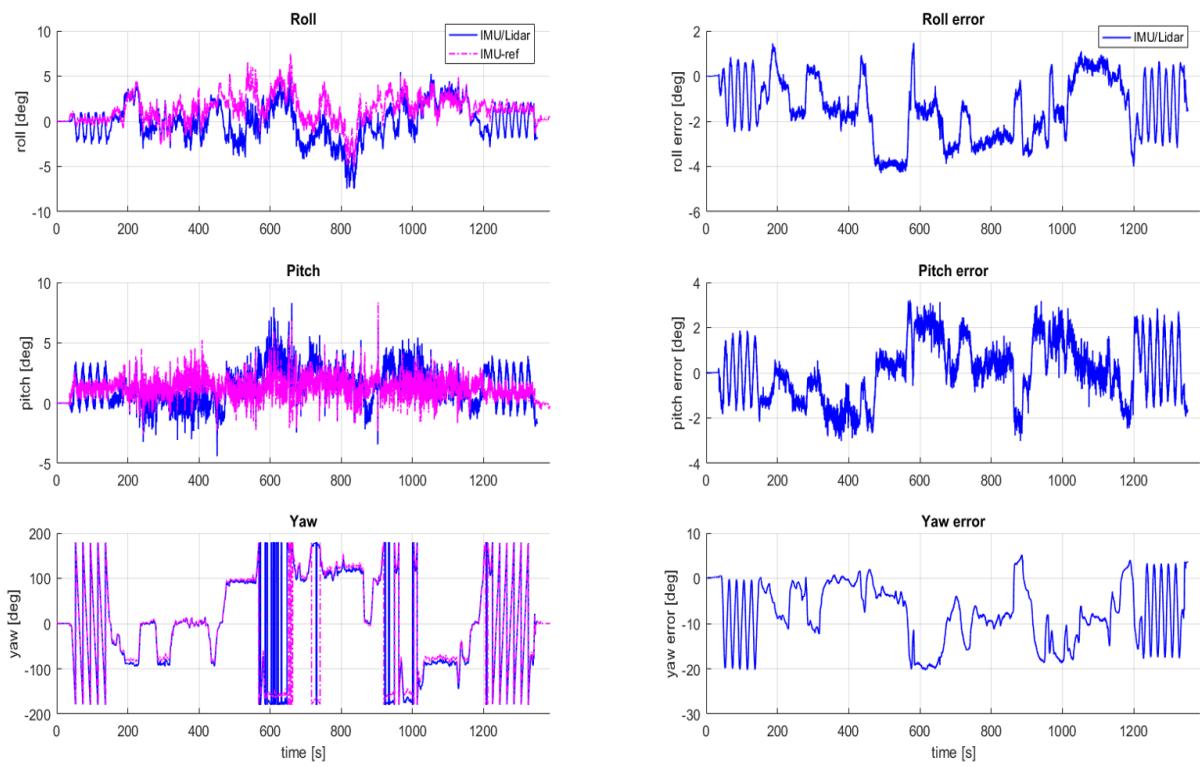
khoảng cách khá xa, RTK sẽ không giữ được độ chính xác do Rover cách quá xa trạm Base do đó dữ liệu RTK trong phần này chỉ mang tính chất tham khảo, cho thấy hình dạng quỹ đạo ước lượng được là hoàn toàn có cơ sở.



Hình 5.38. Dữ liệu vị trí trục z của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT



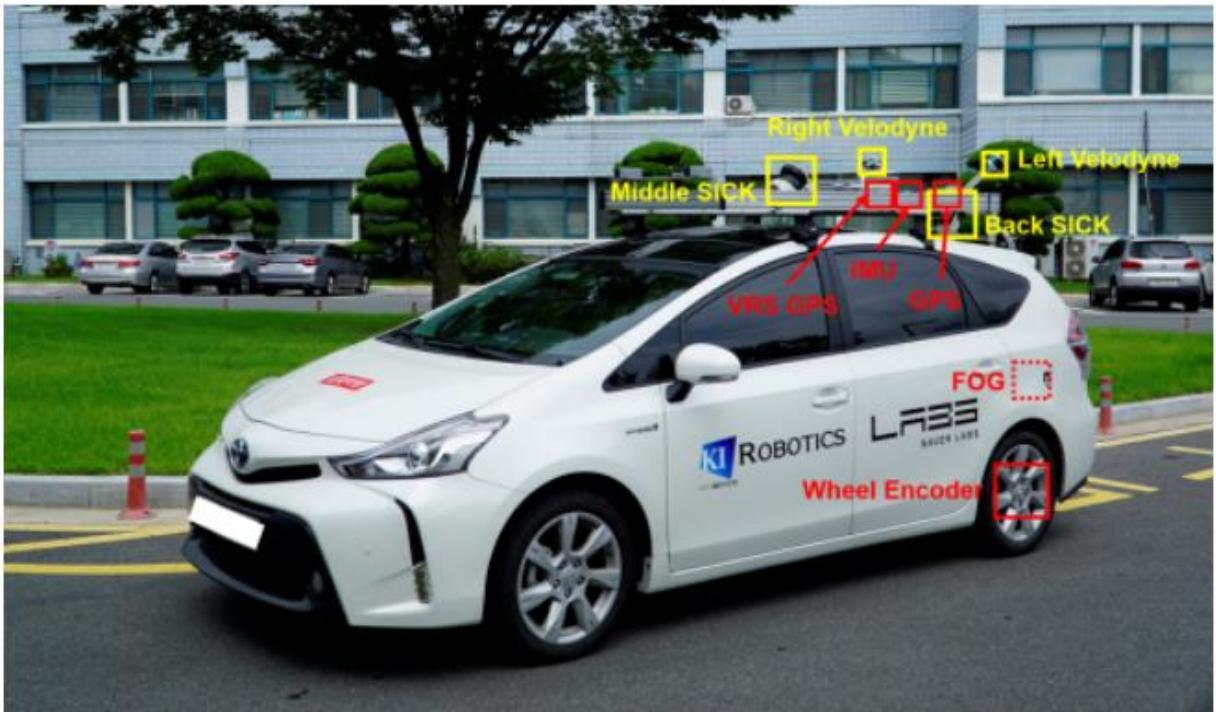
Hình 5.39. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT



Hình 5.40. Dữ liệu góc hướng của IMU/Lidar tập dữ liệu outdoor khuôn viên HCMUT

5.5. Tập dữ liệu có encoder

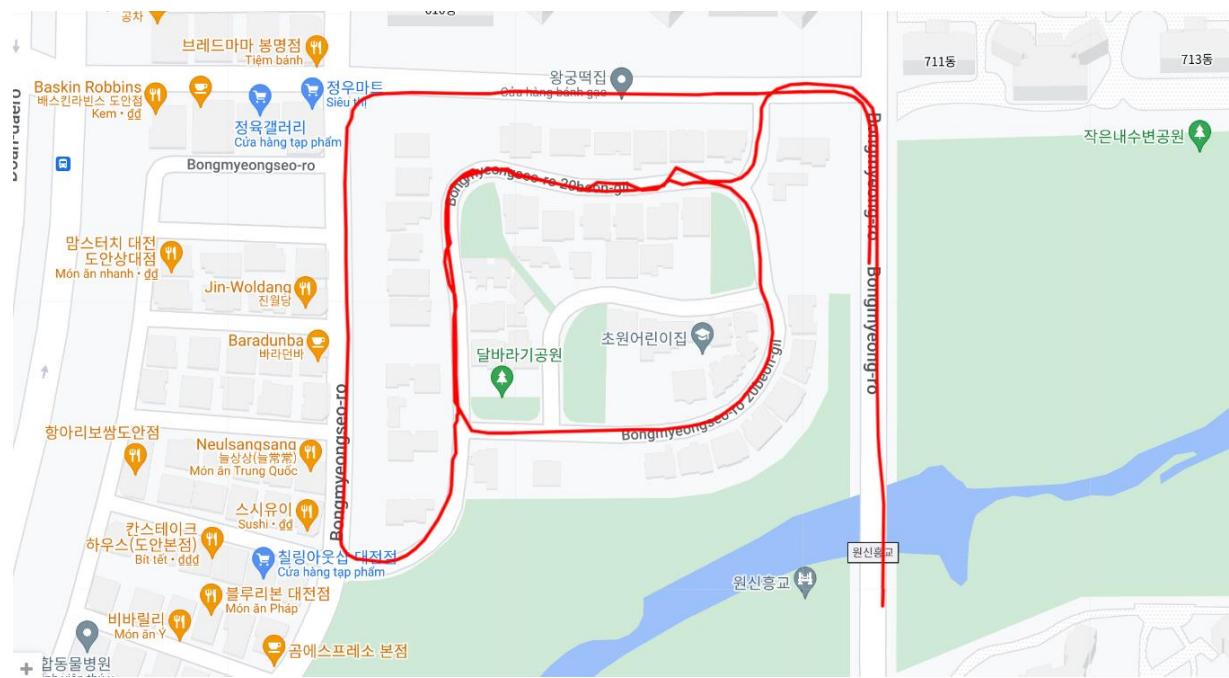
Trong phần này ta sử dụng tập dữ liệu Kaist [52]. Vị trí các cảm biến được bố trí như trong hình bên dưới. Tốc độ cập nhật của IMU, Lidar, Encoder lần lượt là 100Hz, 10Hz và 100Hz. Khu vực khảo sát là một khu dân cư.



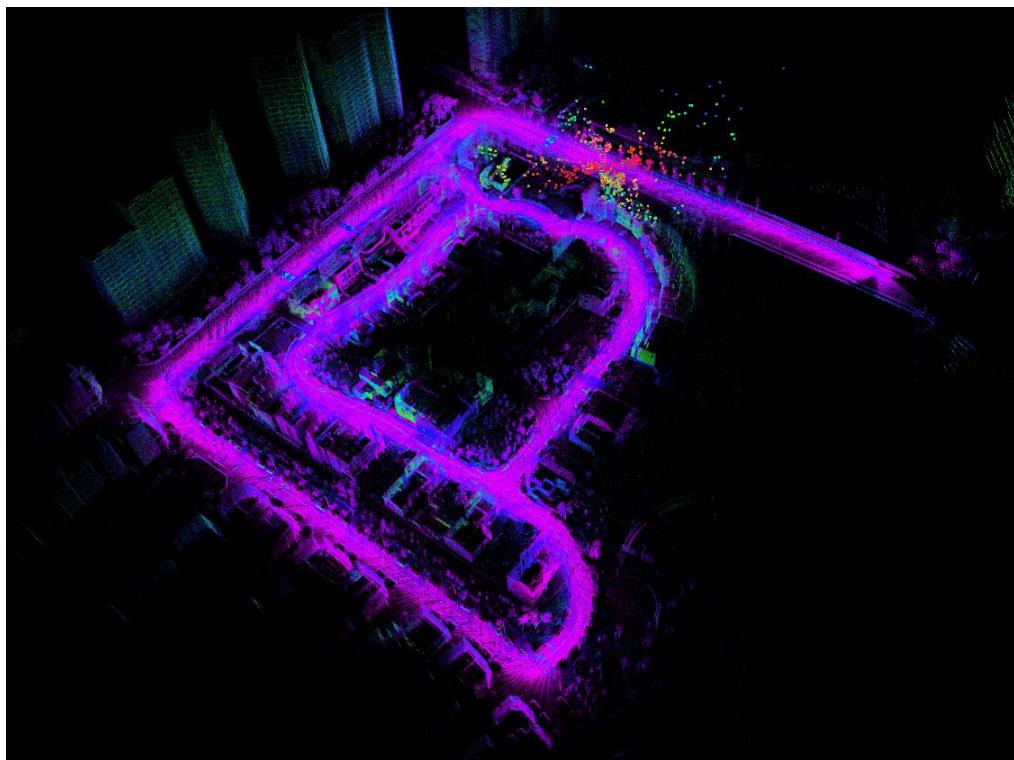
Hình 5.41. Bố trí cảm biến trong tập dữ liệu Kaist [52]

Bảng 5.16. Bảng thông số sử dụng cho các tập dữ liệu Kaist [52]

Ma trận hiệp phương sai Q	$q_g = 0.1, q_a = 0.1,$ $q_{bg} = 0.0001, q_{ba} = 0.0001$
Ma trận hiệp phương sai R (Lidar)	$R = 0.001I_{n \times n}$
Ma trận hiệp phương sai R (Encoder)	$R = 0.01I_{3 \times 3}$
Ma trận biến đổi thuận nhất từ L-frame sang I-frame ${}^I T_L$	${}^I R_L = \begin{bmatrix} -0.515105 & -0.702383 & -0.491249 \\ 0.487008 & -0.711468 & 0.506593 \\ -0.70533 & 0.0217062 & 0.708547 \end{bmatrix}$ ${}^I p_L = [-0.368343 \quad 0.395882 \quad 0.24095]^T$



Hình 5.42. Vị trí của khu vực khảo sát [52]



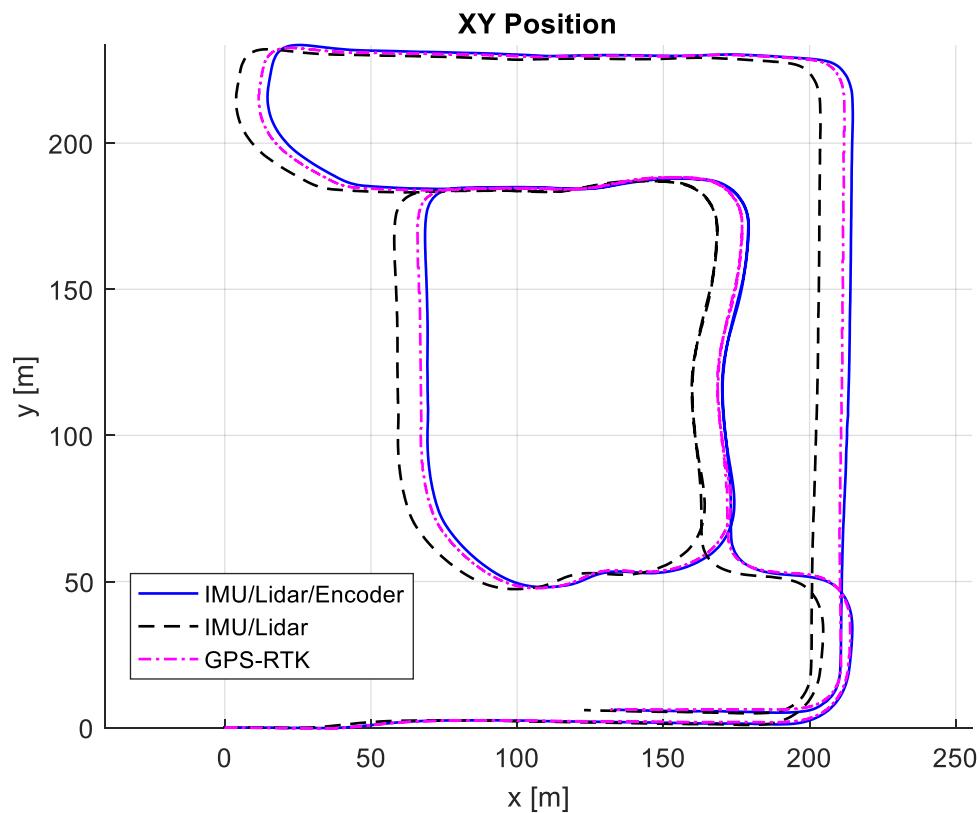
Hình 5.43. Bản đồ dựng được tại khu vực khảo sát

Bảng 5.17. Thông tin tập dữ liệu Kaist

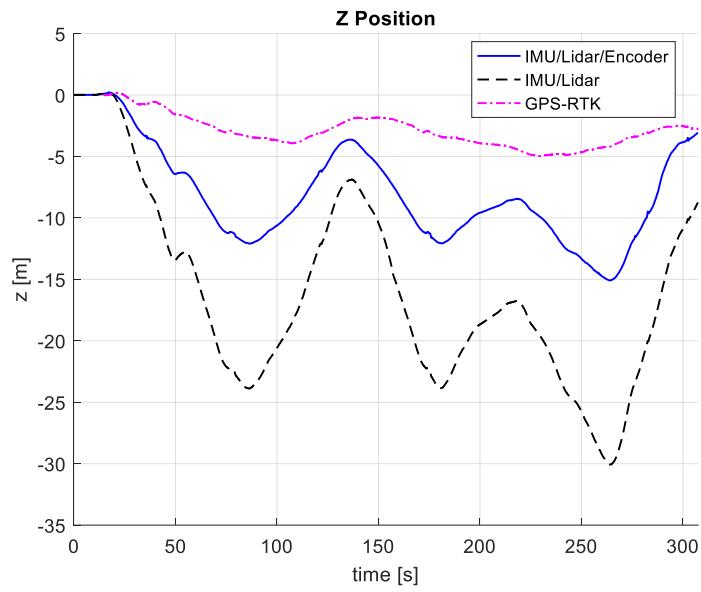
Tổng quãng đường	Thời gian di chuyển	Vận tốc trung bình
1554.631 m	307.355 s	5.058 m/s

Bảng 5.18. Kết quả thực nghiệm tập dữ liệu Kaist

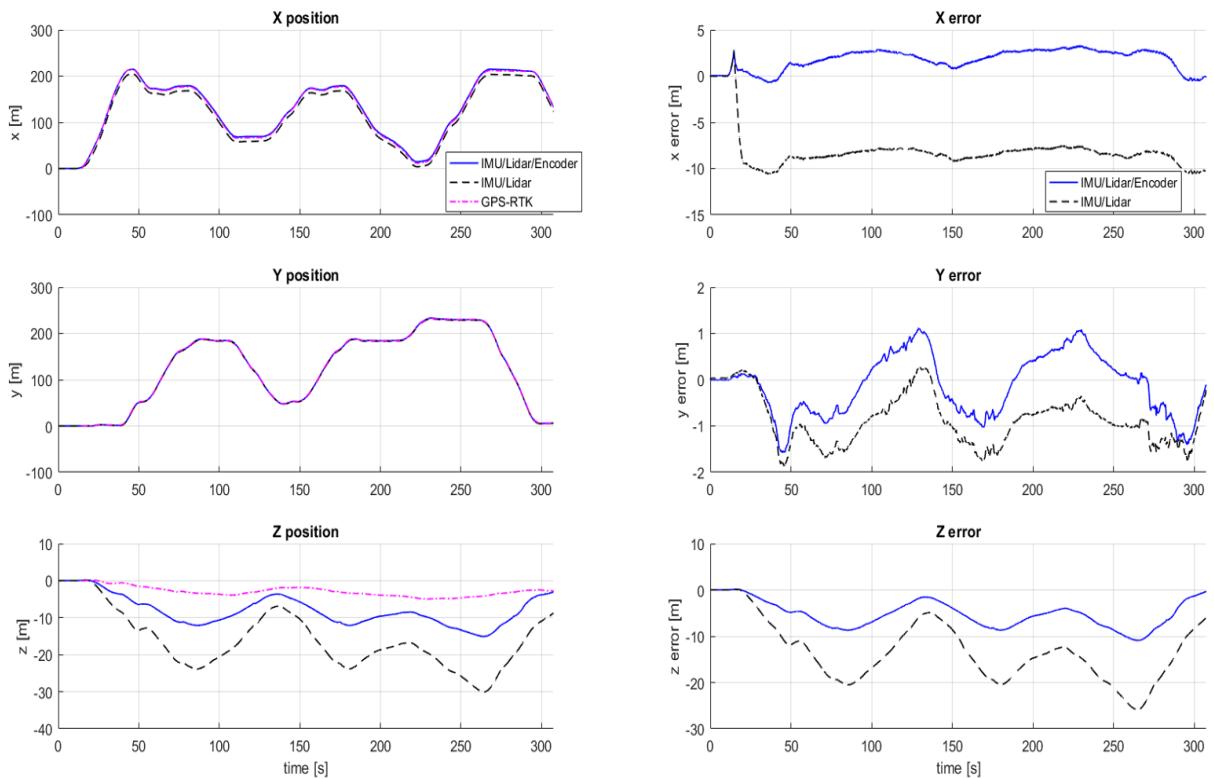
	IMU/Lidar/Encoder	IMU/Lidar
Sai số RMS	2.189 m	8.440 m
Phản trăm sai số theo quãng đường tại thời điểm kết thúc	0.012 %	0.662 %



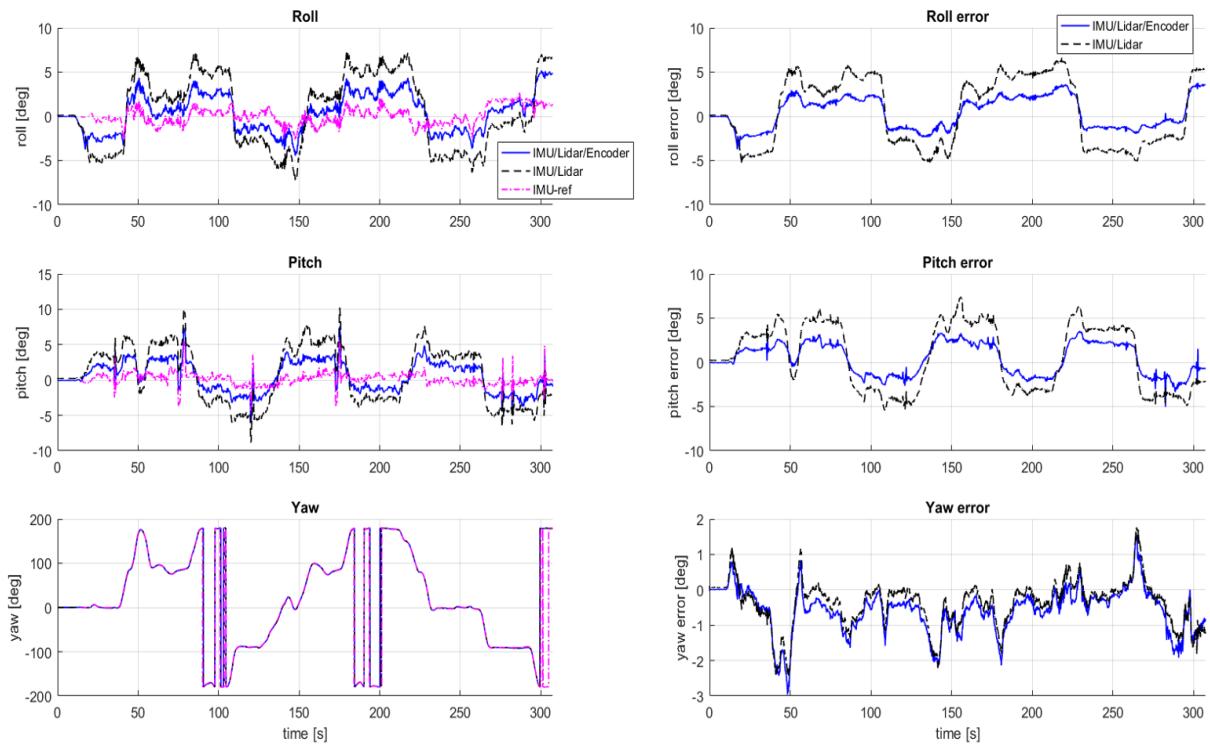
Hình 5.44. Quỹ đạo 2D tập dữ liệu Kaist



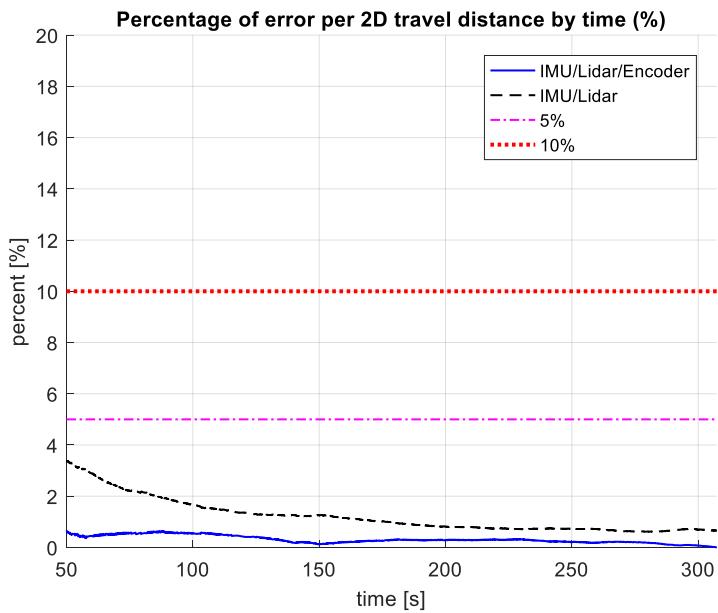
Hình 5.45. Dữ liệu vị trí trục z của IMU/Lidar/Encoder và IMU/Lidar tập dữ Kaist



Hình 5.46. So sánh dữ liệu vị trí trục x, y, z của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu Kaist



Hình 5.47. Dữ liệu góc hướng của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu outdoor Kaist



Hình 5.48. Phần trăm sai số theo quãng đường của IMU/Lidar/Encoder và IMU/Lidar tập dữ liệu Kaist

5.6. Nhận xét

Từ các thí nghiệm trên có thể thấy hệ IMU/Lidar có độ chính xác và ổn định cao hơn IMU/Camera do đặc tính Camera chịu ảnh hưởng bởi nhiều yếu tố như cường độ sáng, độ sâu của ảnh, việc so sánh, matching giữa các điểm diễn ra khó khăn hơn. Bên cạnh đó vị trí trục z của hệ IMU/Lidar mặc dù có độ chính xác cao hơn hệ IMU/Camera nhưng vẫn còn bị trôi. Ngoài ra giải thuật tích hợp IMU/Lidar sử dụng bộ lọc Kalman lặp vẫn giữ được độ chính xác khi môi trường xung quanh có vật thể di chuyển.

Đặc biệt thí nghiệm với tập dữ liệu Kaist cho thấy sự hiệu quả khi kết hợp hệ IMU/Lidar với Encoder, giúp cải thiện độ chính xác vị trí. Tất cả các tập dữ liệu bao gồm tập dữ liệu thu thập được ở trường Đại học Bách Khoa TP.HCM và tập dữ liệu Kaist đều cho phần trăm sai số theo quãng được tại thời điểm kết thúc nhỏ hơn 1%.

Chương 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Luận văn đã hoàn thành mục tiêu đề ra, xây dựng bộ định vị tích hợp IMU/Lidar sử dụng bộ lọc Kalman lặp với phần trăm sai số theo quãng đường nhỏ hơn 1%, tần số cập nhật 10 Hz (bằng với tần số cập nhật của Lidar). Bên cạnh đó luận văn phát triển giải thuật tích hợp IMU/Lidar/Encoder giúp cải thiện độ chính xác vị trí.

Hệ thống phần cứng và phần mềm được xây dựng để thu thập dữ liệu và đánh giá hệ thống. Các dữ liệu thu thập được bao gồm các cảm biến IMU, Lidar, Camera, GPS, RTK có thể được tận dụng lại để thử nghiệm với các giải thuật khác hoặc dùng để đánh giá các giải thuật mới.

Nhiều thử nghiệm đã được thực hiện với các địa hình, khu vực, tốc độ di chuyển, điều kiện môi trường khác nhau để chứng minh sự hiệu quả của giải thuật. Mặc dù giải thuật tích hợp IMU/Lidar chưa tích hợp các thiết bị sử dụng vị trí global như GPS nhưng độ chính xác trong không gian 2D vẫn được duy trì trong khoảng thời gian dài so với hệ IMU/Camera.

6.2. Hướng phát triển

- Khi di chuyển trong một khu vực nhất định, phương tiện tự hành có thể sẽ đi qua một vị trí nhiều lần, vì vậy có thể sử dụng phương pháp loop closer giúp bộ định vị phát hiện được những nơi đã đi qua, từ đó cải thiện độ chính xác vị trí. [8], [61]
- Như đã biết, kết quả vị trí trực z chưa được tốt lắm, vì vậy ta có thể kết hợp dữ liệu từ altimeter để cải thiện độ chính xác vị trí trực z.
- Mặc dù luận văn không đánh giá sai số góc hướng nhưng ta vẫn có thể kết hợp giải thuật IMU/Lidar với giải thuật ước lượng góc riêng biệt [62] để cải thiện độ chính xác của góc hướng, từ đó cải thiện độ chính xác vị trí.

- Kết hợp dữ liệu encoder của vô lăng bánh lái cải thiện độ chính xác góc yaw, từ đó cải thiện độ chính xác vị trí. [63]
- Thay đổi cấu trúc tổ chức phần mềm thành đa luồng để tăng tốc độ xử lý, tính toán. [8]

TÀI LIỆU THAM KHẢO

- [1] Tagliabue, A., Tordesillas, J., Cai, X., Santamaria-Navarro, A., How, J. P., Carlone, L., & Agha-mohammadi, A. A., "LION: Lidar-Inertial observability-aware navigator for Vision-Denied environments," *International Symposium on Experimental Robotics (Springer, Cham)*, pp. 380-390, 2020.
- [2] Gao, Y., Liu, S., Atia, M. M., & Noureldin, A, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors*, pp. 23286-23302, 2015.
- [3] Zhang, J., & Singh, S., "Loam: Lidar odometry and mapping in realtime," *Robotics: Science and Systems*, vol. 2, 2014.
- [4] Segal, A., Haehnel, D., & Thrun, S., "Generalized-ICP," *Robotics: science and systems*, vol. 2, p. 435, 2009.
- [5] Shan, T., & Englot, B., "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4758-4765, 2018.
- [6] Lin, J., & Zhang, F., "Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3126-3131, 2020.
- [7] Xu, W., & Zhang, F., "Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robotics and Automation Letters*, pp. 3317-3324, 2021.
- [8] Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D., "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, 2020.
- [9] Li, K., Li, M., & Hanebeck, U. D., "Towards high-performance solid-state-lidar-inertial odometry and mapping," *IEEE Robotics and Automation Letters*, pp. 5167-5174, 2021.
- [10] Zhang, J., Wen, W., Huang, F., Chen, X., & Hsu, L. T., "Coarse-to-Fine Loosely-Coupled LiDAR-Inertial Odometry for Urban Positioning and Mapping," *Remote Sensing*, p. 2371, 2021.
- [11] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," *IEEE International Conference on Robotics and Automation (ICRA)*, p. 6240–6245, 2017.

- [12] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis, "A laser-aided inertial navigation system (l-ins) for human localization in unknown indoor environments," *IEEE International Conference on Robotics and Automation*, p. 5376–5382, 2010.
- [13] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," *IEEE International Conference on Robotics and Automation (ICRA)*, p. 8899–8906, 2020.
- [14] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," *IEEE International Conference on Robotics and Automation (ICRA)*, p. 3144–3150, 2019.
- [15] Chen, B., Zhao, H., Zhu, R., & Hu, Y, "Marked-LIEO: Visual Marker-Aided LiDAR/IMU/Encoder Integrated Odometry," *Sensors*, 2022.
- [16] Yuan, Zikang, et al, "LIW-OAM: Lidar-Inertial-Wheel Odometry and Mapping," *arXiv*, 2023.
- [17] Filip, I., Pyo, J., Lee, M., & Joe, H, " LiDAR SLAM with a Wheel Encoder in a Featureless Tunnel Environment," *Electronics*, 2023.
- [18] "BISinfotech," [Online]. Available: <https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fwww.bisinfotech.com%2Fanalogs-devices-new-smallest-footprint-precision-tactical-grade-mems-imu-makes-breakthrough-in-positioning-and-navigation-applications%2F&psig=AOvVaw3XzeRIIOU5E2Rlg5DOWkE9&ust=17035>.
- [19] "Mouser," [Online]. Available: <https://www.mouser.fr/images/marketingid/2020/img/124280951.png?v=070223.0507>.
- [20] C. L. Manh, *IMU 9-DOF Calibration Method and Euler Angle Estimator*, 2018.
- [21] Techulator. [Online]. Available: <https://www.techulator.com/resources/8930-How-does-smart-phone-accelerometer-work.aspx>.
- [22] "Physics," [Online]. Available: <https://physics.stackexchange.com/questions/694799/why-are-there-so-many-fundamental-differences-between-contact-forces-and-gravity#:~:text=Accelerometers%20are%20only%20able%20to,the%20body%20being%20properly%20accelerated..>
- [23] "Vatlypt," [Online]. Available: <https://vatlypt.com/luc-va-phan-luc-vat-li-lop-10.html>.

- [24] "Wikipedia," [Online]. Available: https://vi.wikipedia.org/wiki/Hi%20%28u%29ng_Coriolis.
- [25] "Analog IC Tips," [Online]. Available: <https://www.analogictips.com/gyroscopes-part-2-optical-and-mems-implementations-faq/>.
- [26] "Hexagon," [Online]. Available: <https://novatel.com/solutions/attitude>.
- [27] "Vinfastauto," [Online]. Available: https://vinfastauto.com/vn_vi/tim-hieu-cam-bien-lidar-va-ung-dung-tren-o-to.
- [28] "thegioididong," [Online]. Available: <https://www.thegioididong.com/hoi-dap/video-cach-ghi-nguon-trich-dan-va-tao-danh-muc-trich-dan-1382224>.
- [29] "Velodyne VLP-16 Puck," [Online]. Available: <https://airsupply.com/product/velodyne-vlp-16-puck-lite-2/>.
- [30] "Microcontroller Tips," [Online]. Available: https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fwww.microcontrollertips.com%2Flidar-and-time-of-flight-part-2-operation%2F&psig=AOvVaw272TikDDwI8HaOLGhctHMi&ust=1703674572816000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCJCm_r_4rIMDFQAAAAAdA.
- [31] "MathWorks," [Online]. Available: <https://www.mathworks.com/help/lidar/ug/lidar-processing-overview.html>.
- [32] "Siemens Communities," [Online]. Available: <https://community.sw.siemens.com/servlet/rtaImage?eid=ka64O000000kD3b&feoid=00N4O000006Yxpf&refid=0EM4O0000010ELG>.
- [33] "DF Robot," [Online]. Available: <https://www.dfrobot.com/blog-1643.html>.
- [34] Schott. [Online]. Available: <https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fwww.schott.com%2Fen-dk%2Fproducts%2Fhermetic-packages-for-lidar-sensors-p1000281%2Fapplications&psig=AOvVaw3xcT0qpg9bV5P36a-678nK&ust=1703730668269000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoT>.
- [35] "Velodyne Lidar," [Online]. Available: <https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fvelodynelidar.com%2Fpress-release%2Fvelodyne-lidar-nikon-announce-manufacturing-agreement-for-mass-production-of-velodyne-lidar->

sensors%2F&psig=AOvVaw0jregrPQDLky8bf1HOYjei&ust=1703737954102000&source=i.

- [36] Qingyong Hu, Bo Yang*, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni and Andrew Markham, "Learning Semantic Segmentation of Large-Scale Point Clouds with Random Sampling," *Atex*, 2021.
- [37] "Schott," [Online]. Available: <https://www.google.com.vn/url?sa=i&url=https%3A%2Fwww.schott.com%2Fen-dk%2Fproducts%2Fhermetic-packages-for-lidar-sensors-p1000281%2Fapplications&psig=AOvVaw3xcT0qpg9bV5P36a-678nK&ust=1703730668269000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoT>.
- [38] "Livox Wiki," [Online]. Available: https://livox-wiki-en.readthedocs.io/en/latest/_images/Mid-100_point_cloud_picture.png.
- [39] "Livox Wiki," [Online]. Available: <https://livox-wiki-en.readthedocs.io/en/latest/introduction/production.html>.
- [40] "Han My Viet," [Online]. Available: <https://hanmyviet.vn/bo-ma-hoa-vong-quay-tuong-doi-50mm-loai-truc-autonics-e50s8-1024-3-n-24.html>.
- [41] MECSU. [Online]. Available: <https://mecsu.vn/chi-tiet/bo-phat-xung-encoder-omron-360pr-05m-oms-e6b2-cwz6c.LKyer>.
- [42] "Amazon," [Online]. Available: <https://amazon.com.vn/encoder-la-gi.html>.
- [43] X. Blog. [Online]. Available: <https://xtmechanicalblog.com/post/124/bai-viet-so-5-nguyen-ly-hoat-dong-va-tinh-nang-cua-encoder-tu.html>.
- [44] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles.
- [45] Sola, J., Deray, J., & Atchuthan, D. , "A micro Lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.
- [46] Bloesch, M., Burri, M., Omari, S., Hutter, M., & Siegwart, R, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, pp. 1053-1072, 2017.
- [47] Bell, B. M., & Cathey, F. W., "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Transactions on Automatic Control*, pp. 294-297, 1993.
- [48] Barfoot, T. D., State estimation for robotics, Cambridge University Press, 2017.

- [49] Shin, E. H., & El-Sheimy, N, "Accuracy improvement of low cost INS/GPS for land applications," *Proceedings of the 2002 national technical meeting of the institute of navigation*, pp. 146-157, 2022.
- [50] Xu, W., Cai, Y., He, D., Lin, J., & Zhang, F., "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, 2022.
- [51] Cai, Y., Xu, W., & Zhang, F., " ikd-tree: An incremental kd tree for robotic applications," *arXiv preprint*, 2021.
- [52] Jinyong Jeong and Younggun Cho and Young-Sik Shin and Hyunchul Roh and Ayoung Kim, "Complex Urban Dataset with Multi-level Sensors from Highly Diverse Urban Environments," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 642-657, 2019.
- [53] E.-H. Shin, *Accuracy Improvement of Low Cost INS/GPS for Land Applications*, 2001.
- [54] Tran, N. H., and Vinh-Hao Nguyen, "A tightly-coupled visual-inertial system with synchronized time for indoor localization," *International Journal of Mechanical Engineering and Robotics Research*, pp. 202-208, 2021.
- [55] Analog Devices, "ADIS16488A Datasheet," Analog Devices, 2015.
- [56] "Analog Devices Inc. ADIS16488 Tactical Grade Inertial Sensors," [Online]. Available: <https://www.mouser.vn/new/analog-devices/adi-adis16488-sensors/>.
- [57] Velodyne Lidar, "VLP-16 User Manual," 2019.
- [58] "Bumblebee®2 FireWire," [Online]. Available: <https://www.flir.eu/support/products/bumblebee2-firewire/#Overview>.
- [59] "Bumblebee2," [Online]. Available: <https://voltrium.wordpress.com/machine-vision/home/stereo-products/>.
- [60] South, "S82T User manual".
- [61] Kim, Giseop, Sunwook Choi, and Ayoung Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, pp. 1856-1874, 2021.
- [62] Tien-Dung Quoc Tran and Vinh-Hao Nguyen, "Heading Estimation for Autonomous Robot Using Dual-Antenna GPS," 2020.
- [63] CHEN Xianqin, CHEN Hui, "Wheel-LiDAR Odometry and Mapping for Autonomous Vehicles," 2021.

- [64] Y. Balazadegan Sarvood, S. Hosseinyalamdary, and Y. Gao, "Visuallidar odometry aided by reduced imu," *ISPRS international journal of geo-information*, vol. 5, p. 3, 2016.
- [65] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidarinertial-camera odometry," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 5848–5854, 2019.
- [66] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidarinertial 3d plane slam," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 123–130, 2018.
- [67] C. Forster, L. Carbone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, pp. 1-21, 2016.
- [68] M. Hsiao, E. Westman, and M. Kaess, "Dense planar-inertial slam with structural constraints," *IEEE International Conference on Robotics and Automation (ICRA)*, p. 6521–6528, 2018.
- [69] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in gps-denied environments using onboard sensing," *IEEE International Conference on Robotics and Automation*, p. 1–8, 2012.
- [70] Z. Cheng, D. Liu, Y. Yang, T. Ling, X. Chen, L. Zhang, J. Bai, Y. Shen, L. Miao, and W. Huang, "Practical phase unwrapping of interferometric fringes based on unscented kalman filter technique," *Optics express*, vol. 23, p. 32 337–32 349, 2015.
- [71] Liu, S., Atia, M. M., Gao, Y., Givigi, S., & Noureldin, A., "An inertial-aided lidar scan matching algorithm for multisensor land-based navigation," *Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, pp. 2089-2096, 2014.
- [72] Zhen, W., Zeng, S., & Soberer, S., "Robust localization and localizability estimation with a rotating laser scanner," *IEEE international conference on robotics and automation (ICRA)*, pp. 6240-6245, 2017.
- [73] D. Simon, Optimal state estimation: Kalman, H infinity, and nonlinear approaches, 2006.
- [74] "Analog Devices Inc. ADIS16488 Tactical Grade Inertial Sensors," [Online]. Available: <https://www.mouser.vn/new/analog-devices/adi-adiis16488-sensors/>.

- [75] "Stack Exchange," [Online]. Available: <https://engineering.stackexchange.com/questions/38866/how-is-the-yaw-angle-calculated-by-an-imu-sensor>.
- [76] A. I. SKILLS. [Online]. Available: https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fmaster-engineer.com%2F2020%2F11%2F05%2Fwriting-ros-node%2F&psig=AOvVaw0kUqhVWxIRCy7sIX1MtMqC&ust=1703671783058000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCNC-0Y_urIMDFQAAAAAdAAAAABAv.
- [77] "Granitex," [Online]. Available: <https://www.google.com.vn/url?sa=i&url=https%3A%2F%2Fgranitex.dz%2Fsouth-s82t-k.html&psig=AOvVaw2UXrP78BoLVGPNZ4pfdkXf&ust=1703672853133000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCND75bjzrIMDFQAAAAAdAAAAABAn>.
- [78] V. Lidar. [Online]. Available: <https://velodynelidar.com/blog/guide-to-lidar-wavelengths/>.
- [79] "3D Plus," [Online]. Available: <https://in3dplus.com/cong-nghe-lidar-la-gi/>.
- [80] "Arxiv," [Online]. Available: <https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRN8rkMH7BVMpfkGwTFyVlei38gKUitTRqO16LROTolrf3gJ9UOdYOHxpGNiBoER0AmzN8&usqp=CAU>.
- [81] "LIO-SAM Github," [Online]. Available: <https://github.com/TixiaoShan/LIO-SAM>.
- [82] Chang, Le, Xiaoji Niu, and Tianyi Liu, "GNSS/IMU/ODO/LiDAR-SLAM integrated navigation system using IMU/ODO pre-integration," *Sensors*, 2020.

LÝ LỊCH TRÍCH NGANG CỦA TÁC GIẢ

Họ và tên: Cao Huỳnh Phúc

Ngày, tháng, năm sinh: 11/09/1998

Nơi sinh: TP. Hồ Chí Minh

Địa chỉ liên lạc: 8/2 đường 147, phường Phước Long B, TP. Thủ Đức, TP.HCM

QUÁ TRÌNH ĐÀO TẠO:

- 09/2016 đến 11/2020: học đại học chuyên ngành Kỹ thuật Điều khiển và Tự động hóa, trường Đại học Bách Khoa, ĐHQG TP.HCM
- 09/2020 đến nay: học cao học chuyên ngành Kỹ thuật Điều khiển và Tự động hóa, trường Đại học Bách Khoa, ĐHQG TP.HCM

QUÁ TRÌNH CÔNG TÁC:

- 06/2021 đến 10/2022: Kỹ sư phần mềm nhúng tại công ty TNHH Bosch Global Software Technologies, 364 Cộng Hòa, quận Tân Bình, TP. Hồ Chí Minh
- 11/2022 đến nay: Kỹ sư kiểm định vi mạch tại công ty TNHH Ampere Computing, lô D.01 KCX Tân Thuận, Tân Thuận Đông, quận 7, TP. Hồ Chí Minh