

ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Trần Thuận Hoàng

**NGHIÊN CỨU PHƯƠNG PHÁP
TỔNG HỢP CẢM BIẾN
DÙNG CHO KỸ THUẬT DẪN ĐƯỜNG
CÁC ROBOT DI ĐỘNG**

Chuyên ngành: Kỹ thuật Điện tử
Mã số chuyên ngành: 62 52 70 01

LUẬN ÁN TIẾN SĨ NGÀNH CÔNG NGHỆ ĐIỆN TỬ VIỄN THÔNG

NGƯỜI HƯỚNG DẪN KHOA HỌC:
1. PGS. TS. TRẦN QUANG VINH
2. PGS. TS. BẠCH GIA DƯƠNG

Hà Nội - 2015

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là công trình nghiên cứu của bản thân tác giả. Các kết quả nghiên cứu và các kết luận trong luận án này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu (nếu có) đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả luận án

MỤC LỤC

	Trang
Trang phụ bìa	i
Lời cam đoan.....	ii
Mục lục	iii
Danh mục các ký hiệu và chữ viết tắt	vi
Danh mục các bảng	vii
Danh mục các hình vẽ, đồ thị.....	vii
MỞ ĐẦU	1
CHƯƠNG 1. PHƯƠNG PHÁP TỔNG HỢP CẢM BIẾN	10
1.1. Hệ thống tổng hợp dữ liệu cảm biến	12
1.1.1. Tổng hợp dữ liệu đa cảm biến.....	12
1.1.2. Quản lý đa cảm biến	17
1.1.3. Phân loại tổng hợp đa cảm biến	19
1.1.3.1. Phân loại theo mức	19
1.1.3.2. Phân loại dựa trên dữ liệu vào ra	20
1.1.3.3. Phân loại dựa trên cấu hình cảm biến	21
1.1.4. Các vấn đề cần giải quyết trong bài toán tổng hợp dữ liệu đa cảm biến ...	23
1.1.5. Kiến trúc hệ thống tổng hợp dữ liệu	26
1.1.5.1. Kiến trúc tổng hợp trung tâm	26
1.1.5.2. Kiến trúc tổng hợp phân tán	26
1.1.5.3. Kiến trúc tổng hợp cục bộ	27
1.2. Các phương pháp tổng hợp cảm biến cho robot di động.....	28
1.2.1. Định vị và lập bản đồ.....	28
1.2.2. Tổng hợp cảm biến với phương pháp suy luận xác suất để nâng cao độ tin cậy cho bài toán định vị robot	33
1.2.3. Tổng hợp cảm biến bằng bộ lọc Kalman để nâng cao độ tin cậy cho bài toán định vị robot.	34
1.3. Kết luận	40
CHƯƠNG 2. XÂY DỰNG MÔ HÌNH ROBOT DI ĐỘNG ĐA CẢM BIẾN	41
2.1. Thiết kế chế tạo phần cứng mô hình robot di động đa cảm biến.	41
2.1.1. Các mô đun cảm nhận, khối điều khiển và cơ cấu chấp hành của robot....	41
2.1.1.1. Cấu trúc cơ khí của robot.....	43
2.1.1.2. Các mô đun điều khiển chuyển động.	44

2.1.1.3. Các mô đun cảm biến trên robot.....	44
2.1.1.4. Khối điều khiển điện tử trong robot.....	53
2.1.1.5. Các khối giao tiếp truyền tin giữa vi điều khiển MCUdsPIC với máy tính	56
2.1.2. Chương trình điều khiển hệ thống	57
2.2. Đo đạc đánh giá mô hình hệ thống được chế tạo	59
2.2.1. Kiểm tra độ chính xác của chuyển động robot	59
2.2.1. Kiểm tra độ tin cậy của ảnh laser.....	60
2.3. Kết luận.....	63

CHƯƠNG 3. TỔNG HỢP CẢM BIẾN DÙNG CHO ĐỊNH VỊ VÀ LẬP BẢN ĐỒ DẪN ĐƯỜNG ROBOT DI ĐỘNG

CHƯƠNG 3. TỔNG HỢP CẢM BIẾN DÙNG CHO ĐỊNH VỊ VÀ LẬP BẢN ĐỒ DẪN ĐƯỜNG ROBOT DI ĐỘNG	64
3.1. Tổng hợp cảm biến bằng bộ lọc Kalman mở rộng để nâng cao độ tin cậy của phép định vị robot	65
3.1.1. Tổng hợp dữ liệu odometry với cảm biến chỉ hướng từ-địa bàn.....	70
3.1.1.1. Xác định các ma trận trong các bước tổng hợp dùng EKF	70
3.1.1.2. Thực nghiệm và thảo luận	72
3.1.2. Tổng hợp dữ liệu với thêm cảm biến đo xa laser.....	73
3.1.2.1. Tính véc-tơ z_k , ma trận H_k và ma trận hiệp phương sai R_k	74
3.1.2.2. Nhận dạng các tham số $\text{var}(r_i)$ và $\text{var}(\psi_i)$ của ma trận R_k	77
3.1.2.3. Thực nghiệm và thảo luận	81
3.1.3. Tổng hợp dữ liệu với cảm biến camera ảnh toàn phương	84
3.1.3.1. Tính véc-tơ z_k , ma trận H_k và ma trận hiệp phương sai R_k	84
3.1.3.2. Áp dụng kỹ thuật thị giác máy tính phát hiện cột mốc và đo góc γ_k ..	87
3.1.3.3. Khảo sát sai số định góc của cảm biến ảnh toàn phương.....	88
3.1.3.4. Kết quả thực nghiệm và thảo luận	89
3.2. Xây dựng bản đồ dẫn đường bằng tổng hợp dữ liệu cảm biến đo xa laser	91
3.2.1. Thu thập dữ liệu 3D các điểm ảnh của vật cản	91
3.2.2. Giải thuật IPaBD xây dựng bản đồ dẫn đường 2D	92
3.2.3. Thực nghiệm xây dựng bản đồ và áp dụng cho vạch đường đi và điều khiển robot.....	95
3.2.3.1. Kết quả xây dựng bản đồ bằng giải thuật IPaBD	95
3.2.3.2. Thực nghiệm vạch đường đi và điều khiển robot nhờ bản đồ dẫn đường	97
3.3. Kết luận.....	99

CHƯƠNG 4. ĐIỀU KHIỂN CHUYỂN ĐỘNG	100
4.1. Sự ổn định Lyapunov và nguyên lý bất biến LaSalle	100
4.1.1. Khái niệm ổn định Lyapunov đối với hệ tự trị	100
4.1.2. Nguyên lý bất biến LaSalle.....	102
4.2. Xây dựng bộ điều khiển chuyển động ổn định	102
4.2.1. Mô hình động học.....	103
4.2.2. Thiết kế bộ điều khiển	106
4.2.2.1. Điều khiển ổn định trong tập cấu hình toàn cục Ω_G	106
4.2.2.2. Điều khiển ổn định hệ thống trong tập cấu hình cục bộ Ω_L	109
4.2.3. Sử dụng bộ lọc Kalman cho vòng điều khiển phản hồi	111
4.2.4. Mô phỏng và thực nghiệm.....	112
4.3. Kết luận.....	116
KẾT LUẬN VÀ THẢO LUẬN HƯỚNG PHÁT TRIỂN	117
1. Các nội dung được thực hiện trong Luận án.....	117
2. Những đóng góp chính của Luận án	118
3. Hướng phát triển tương lai.....	118
DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN	120
TÀI LIỆU THAM KHẢO	122

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

I. Các ký hiệu

β	góc chệch hướng của xung laser so với trục X
α	góc ngảng của xung laser với mặt phẳng (x,y)
r	khoảng cách đo được của cảm biến laser
γ	góc hướng của robot và cột mốc trong phép đo ảnh toàn phương
θ	góc hướng của robot, lệch so với trục X
Δt	chu kỳ lấy mẫu số liệu đo
$\omega_R; \omega_L$	vận tốc góc của hai bánh xe phải và trái
\mathbf{x}	véc-tơ trạng thái của robot
\mathbf{w}	véc-tơ nhiễu hệ thống
\mathbf{z}	véc-tơ đo lường (véc-tơ số đo)
\mathbf{v}	véc-tơ nhiễu đo
\mathbf{u}	véc-tơ điều khiển lối vào
P	ma trận hiệp phương sai của sai số dự báo trạng thái
Q	ma trận hiệp phương sai của nhiễu hệ thống
R	ma trận hiệp phương sai của nhiễu đo
K	hệ số (độ lợi) lọc Kalman
A	ma trận đạo hàm riêng của hàm hệ thống f theo \mathbf{x}
W	ma trận đạo hàm riêng của hàm hệ thống f theo \mathbf{w}
H	ma trận đạo hàm riêng của hàm đo h theo \mathbf{x}
V	ma trận đạo hàm riêng của hàm đo h theo \mathbf{v}
I	ma trận đơn vị

II. Các chữ viết tắt

AI	trí tuệ nhân tạo (Artificial Intelligence)
AGV	xe vận tải tự động (Autonomous Guided Vehicles)
LRF	cảm biến đo xa laser (Laser Range Finder)

EKF	bộ lọc Kalman mở rộng (Extended Kalman Filter)
GPS	hệ định vị toàn cầu (Global Positioning System)
PID	bộ điều khiển tỉ lệ vi tích phân (Proportional-Integral-Derivative)
GYRO	con quay (Gyroscope)
INS	hệ dẫn đường quán tính (Inertial Navigation System)
INS/GPS	hệ tích hợp INS và GPS
KF	bộ lọc Kalman (Kalman Filter)
PFM	phương pháp trường thế (Potential Field Methods)
VFF	trường lực ảo (Virtual Force Field)
VFH	tọa độ cực (Virtual Field Histogram)
PWM	điều chế độ rộng xung (Pulsed Width Modulation)
UART	mô-đun truyền thông nối tiếp (Universal Asynchronous Receiver/Transmitter)
CAN	mạng khu vực điều khiển (Control Area Network)
ADC	chuyển đổi tín hiệu tương tự - số (Analog-to-Digital Converter)
FPGA	mạng cổng có thể lập trình được (Field-Programmable Gate Array)
IpaBD	ép ảnh và phát hiện vật cản (Image Pressure and Barriers Detection)

DANH MỤC CÁC BẢNG

Bảng 2.1: Các thông số ứng với 2 thời gian quét dọc T_v ngắn và dài.	61
Bảng 3.1: Giá trị của R_k của LRF	82
Bảng 3.2: Các mẫu phép đo của cảm biến camera toàn phương.	89
Bảng 3.3: Sai số căn quân phương RMSE của các cấu hình tổng hợp cảm biến so với đường thực	90

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

Hình 1.1: Sơ đồ khối của: a) tổng hợp đa cảm biến; b) tích hợp đa cảm biến	14
Hình 1.2: Quá trình xử lý thông minh của con người	15
Hình 1.3: Hệ thống quản lý đa cảm biến.	18
Hình 1.4: Các đặc điểm tổng hợp khác nhau dựa trên đầu vào /ra	21
Hình 1.5: Tổng hợp cấu hình: cạnh tranh, bổ sung và cộng tác	22
Hình 1.6: Kiến trúc trung tâm với một trung tâm xử lý	26
Hình 1.7: Kiến trúc tổng hợp phân tán.	27
Hình 1.8: Kiến trúc tổng hợp cục bộ.....	28
Hình 1.9: Sự trôi phép định vị được phát sinh bởi robot Nomad 200	29
Hình 1.10: Tổng hợp đa cảm biến sử dụng kỹ thuật Bayesian.....	34
Hình 1.11: Sơ đồ thuật toán bộ lọc Kalman rời rạc.	36
Hình 1.12: Sơ đồ thuật toán bộ lọc Kalman mở rộng.	39
Hình 2.1: Mạng thông tin cảm nhận trong robot đa cảm biến.....	42
Hình 2.2: Cấu tạo của robot đa cảm biến.	43
Hình 2.3: Mô-đun Motion Mind	44
Hình 2.4: Cảm biến lập mã quang.....	44
Hình 2.5: a) Mô-đun cảm biến từ-địa bàn CMPS03; b) Nguyên lý hoạt động	45
Hình 2.6: a) Mô-đun cảm biến siêu âm; b) Nguyên lý hoạt động	46
Hình 2.7: Cảm biến ảnh camera toàn phương	47
Hình 2.8: a) Dải góc quét 180^0 của LRF; b) Mặt phẳng tia laser với các góc quét β	47
Hình 2.9: Lưu đồ chương trình thu và xử lý số liệu.....	49
Hình 2.10: a) Quét ngẩng lên và xuống cùng không gian hình cầu các điểm đo; b) Cơ cấu truyền động quay và mô-tơ servo.....	50
Hình 2.11: Xác định tọa độ của một điểm ảnh 3D.....	51
Hình 2.12: Chi tiết bản vẽ hệ thống cơ khí tạo quét ngẩng lên-xuống	51
Hình 2.13: Tốc độ mô-tơ khi không và có điều khiển PID	52
Hình 2.14: Góc quét dọc α của LRF	52

Hình 2.15: Mạch in và ảnh chụp mạch lắp ráp vi điều khiển MCUdsPIC.....	54
Hình 2.16: Sơ đồ nguyên lý của mạch điện tử điều khiển dùng MCUdsPIC30F4011	55
Hình 2.17: Sơ đồ nguyên lý của mạch chuyển đổi USB ↔ RS485	56
Hình 2.18: Sơ đồ mạch in của mạch chuyển đổi USB ↔ RS485.....	56
Hình 2.19: Ảnh chụp mạch lắp ráp mạch chuyển đổi USB ↔ RS485	57
Hình 2.20: Lưu đồ quy trình dẫn đường cho robot di động	58
Hình 2.21: a) Robot chạy thẳng 7m; b) Robot chạy và quay với các vận tốc khác nhau	59
Hình 2.22: a) Ảnh camera; b) Ảnh laser 2D; c) Ảnh laser 3D	60
Hình 2.23: a) Các cặp giá trị đo (đường đứt nét) tọa độ y và z của vật và giá trị thực (đường liền nét) phụ thuộc vào góc nghiêng; b) Sai lệch tuyệt đối của giá trị z.....	61
Hình 2.24: Ảnh 3D một vật hình tròn đặt trước LRF	61
Hình 2.25: a) Ảnh camera; b) Ảnh Laser 3D.....	62
Hình 2.26: Ảnh 3D một người đi ngang qua LRF	62
Hình 2.27: Ảnh 3D một người đứng tại chỗ trong khi robot di chuyển về phía đó với vận tốc 0,3 m/s.....	62
Hình 3.1: Các khâu điều khiển trong quá trình dẫn đường	64
Hình 3.2: a) Tư thế và các thông số của robot trong hai hệ tọa độ; b) Mô hình chuyển động.....	65
Hình 3.3: Quỹ đạo ước lượng của robot với các cấu hình không có (màu đen) và có EKF (màu xanh lá cây) so với quỹ đạo thực (màu xanh dương).....	73
Hình 3.4: Máy đo LRF đặt trên robot đo 2 thông số của một đường thẳng trong môi trường.	74
Hình 3.5: Các tham số (ρ , ψ) của các phân đoạn đường thẳng với vị trí robot trong hệ tọa độ của robot.	79
Hình 3.6: Các tham số trùng khớp giữa các phân đoạn đường thẳng cục bộ và toàn cục	79
Hình 3.7: a) Trích chọn các đoạn thẳng với ngưỡng cố định; b) Trích chọn các đoạn thẳng với ngưỡng động.	81

Hình 3.8: a) Ảnh chụp liên tiếp robot chuyển động trong môi trường; b) Làm khớp các đoạn thẳng ở bản đồ toàn cục và bản đồ cục bộ.....	82
Hình 3.9: Quỹ đạo ước tính của robot với các cấu hình EKF khác nhau.....	83
Hình 3.10: Độ lệch giữa các vị trí được ước tính với đường thực.....	83
Hình 3.11: Ước lượng góc nhìn $\hat{\gamma}_k$ từ robot đến vật mốc màu đỏ (x_m, y_m) bằng phép đo odometry và cảm biến ảnh toàn phương.	85
Hình 3.12: Ảnh toàn phương và trải ảnh toàn cảnh của camera-omni	87
Hình 3.13: Phát hiện đường thẳng đúng sử dụng thuật toán Hough	87
Hình 3.14: Ảnh chụp liên tiếp robot chuyển động trong môi trường; b) Quỹ đạo ước tính của robot với các cấu hình EKF khác nhau	89
Hình 3.15: Độ lệch giữa các vị trí được ước tính với đường thực.....	90
Hình 3.16: Ép ảnh 3D thành 2D trên mặt phẳng xy.....	93
Hình 3.17: (a) Ví dụ minh họa quét ảnh 3D, (b) hình chiếu các điểm ảnh hướng về cảm biến trên mặt phẳng xy, ứng với một giá trị góc quét β_k có nhiều điểm ảnh (tròn hay tam giác) có các giá trị R khác nhau, (c) mặt cắt đứng của quá trình quét với các điểm ảnh cùng có giá trị quét β_k nhưng có chiều cao khác nhau nên có giá trị R khác nhau.	93
Hình 3.18: Ảnh chụp 3D môi trường toàn cục.	96
Hình 3.19: Bản đồ 2D với toàn bộ điểm ảnh 3D được ép trên mặt phẳng OXY.	96
Hình 3.20: Kết quả bản đồ thu được do giải thuật IPaBD.	96
Hình 3.21: Vạch đường cho robot trên bản đồ IPaBD bằng giải thuật A*.	98
Hình 3.22: Bản đồ Voronoi với đường đi tối ưu qua nút B.....	98
Hình 4.1: Thế và các thông số của robot.....	104
Hình 4.2: Vòng điều khiển phản hồi với bộ lọc Kalman	112
Hình 4.3: Đáp ứng với các luật điều khiển dùng 1 cấu hình (a) và 2 cấu hình (b).	113
Hình 4.4: Kết quả mô phỏng.....	114
Hình 4.5: Kết quả thực nghiệm.....	114
Hình 4.6: a), b) Kết quả điều khiển ổn định không có bộ lọc Kalman	114
Hình 4.7: a), b) Kết quả điều khiển ổn định có bộ lọc Kalman	115

Hình 4.8: Vận tốc góc ω tiệm cận ổn định đến đích khi không dùng bộ lọc EKF (đường màu xanh) và có dùng EKF (đường màu đỏ) 115

MỞ ĐẦU

1. Đặt vấn đề.

Robot là một hệ thống cơ điện tử, chuyển tải thông tin cảm nhận có mục đích thành chuyển động của hệ thống trong môi trường xung quanh. Một hệ thống robot như vậy sẽ bao gồm 3 bộ phận cơ bản: *bộ cảm nhận*, *bộ điều khiển* và *bộ chấp hành*. Bộ phận cảm nhận có nhiệm vụ thu thập thông tin từ môi trường, thông tin này được đưa vào bộ điều khiển để xử lý và xuất ra các lệnh thích hợp quyết định hành vi của hệ thống. Bộ phận điều khiển được coi là “bộ não” của robot, thường bộ phận này là các máy tính hay các bộ xử lý chuyên dụng và bộ nhớ cùng các giao diện ghép nối vào/ra. Bộ chấp hành thực hiện các lệnh ra từ bộ điều khiển để tác động trở lại môi trường hay chính robot.

Từ những năm 60 của thế kỷ trước, bắt đầu bởi những hoạt động thám hiểm vũ trụ, xu thế thông minh hóa robot đã và đang được phát triển rất nhanh. Đã có một sự phân nhánh trên con đường phát triển các hệ thống robot theo hướng phục vụ công nghiệp và các robot có trí tuệ nhân tạo AI (*Artificial Intelligent*) - còn gọi là các robot thông minh. Các *robot công nghiệp* như các tay máy đã được nghiên cứu chế tạo sử dụng rất thành công trong công nghiệp như lắp ráp vật phẩm, sơn, hàn, đóng gói, kiểm chuẩn trên các dây chuyền sản xuất với độ chính xác và tốc độ cao. Tuy nhiên, nếu so sánh với các *robot thông minh*, có cấp độ thông minh khác nhau, thì các robot công nghiệp chỉ có thể được coi là các hệ thống tự động hóa với một vài chương trình được lập đi lập lại mà thôi. Các robot thông minh từ loại có thể nhận dạng để lựa chọn vật phẩm theo màu sắc, hiểu được các lệnh theo ngôn ngữ tự nhiên, tự động tránh được các vật cản, đến các máy đánh cờ tự động có thể thắng được người chơi. Nếu chương trình điều khiển được nạp toàn bộ vào bộ nhớ của robot, được lưu trữ theo kiểu cố định (nonvolatile) để cho phép một khi được khởi động, robot có thể hoạt động độc lập không cần các tác nhân điều khiển khác, thì đó được gọi là *robot hoạt động kiểu tự trị* (autonomous robot). Hoạt động tự trị yêu cầu robot có thể tự điều khiển trong thời gian dài, không có hoặc có rất ít sự tác

động của người điều khiển. Từ việc hạ cánh tàu vũ trụ lên các hành tinh xa xôi cho tới các phương tiện lặn hoạt động sâu trong lòng đại dương đều cần phải có các hệ thống có khả năng đưa ra các quyết định và hành động có kiểm soát một cách độc lập.

Trong những năm gần đây trên thế giới, cùng với việc thông minh hóa robot là nhu cầu phát triển các robot di động đã dẫn đến sự bùng nổ trong nghiên cứu phát triển các hệ thống *robot di động tự trị* (autonomous mobile robot). *Robot di động* là một máy tự động có khả năng di chuyển đến đích trong một môi trường nào đó. Đây là điều khác với các robot công nghiệp (như tay máy có khớp quay và đầu kẹp) được gắn với một không gian cố định. Khởi đầu bằng các xe vận tải tự động AGV (*autonomous guided vehicles*) đã được ứng dụng rất thành công trong công nghiệp, các robot di động thông minh hiện nay vẫn đang trong thời kỳ phát triển và được coi là có tiềm năng trong tương lai gần. Các tiến bộ công nghệ mới trong việc phát triển các thiết bị cảm biến (sensor) và khả năng tính toán của các hệ xử lý đã thúc đẩy mức độ tự trị trong sự vận hành các robot di động. Mặt khác, những đòi hỏi ứng dụng của robot di động trong các môi trường khác như dịch vụ, giải trí, y tế, an ninh, quân sự cũng hứa hẹn có những tiến bộ quan trọng trong quá trình thiết kế và phát triển các hệ thống này.

Có thể tìm thấy nhiều ứng dụng của robot di động thông minh. Một số robot di động tự trị hoặc bán tự trị đã được giới thiệu cho đến nay như: tên lửa hành trình, máy bay không người lái hoạt động khảo sát trong không gian, v.v... Các robot hoạt động trong các môi trường trong nhà biết trước như các robot làm sạch sàn nhà của Lawitzky [75], robot hướng dẫn viên bảo tàng [114], robot chăm sóc người bệnh Evans [46], robot hút bụi,... Ngoài ra, còn có những robot hoạt động ở môi trường hoàn toàn không biết trước như robot khai thác mỏ Makela [81], robot hoạt động dưới lòng đất [103], robot thám hiểm, v.v...

Không kể hoạt động của các bộ phận gắn trên để robot di động, bài toán dẫn đường cho sự di chuyển của nó từ một điểm xuất phát tới đích một cách an toàn, được gọi tắt là “*dẫn đường cho robot di động*”, được coi là chính yếu trong các

ngiên cứu về robot di động hiện nay. Không khác nhiều so với hành vi của con người trong việc dẫn đường, muốn giải quyết được bài toán này thì robot phải tự xác định được vị trí của mình trong môi trường (positioning), xác lập được bản đồ môi trường nếu cần thiết (mapping), vạch ra được quỹ đạo đi tới đích (path planning) và xuất ra cách thức điều khiển đi trên quỹ đạo (path control) và tránh vật cản (obstacle avoidance) trên đường đi. Leonard và Durant-Whyte [76] đã tóm tắt bài toán chung của dẫn đường robot di động là việc trả lời 3 câu hỏi: “*Where am I?* (robot đang ở đâu)”, “*Where am I going?* (robot sẽ đi tới đâu)”, và “*How should I get there?* (robot sẽ đi tới đó như thế nào)”. Để trả lời cho 3 câu hỏi này robot phải: có một mô hình môi trường (đã cho hoặc tự xây dựng); nhận biết và phân tích môi trường; tìm vị trí của nó trong môi trường; lập kế hoạch và điều khiển chuyển động.

Việc nghiên cứu sử dụng các cảm biến và hệ thống phân cứng hiện đại cũng như phát triển các giải thuật phần mềm nhằm tăng độ tin cậy khi giải quyết câu hỏi thứ nhất: *định vị robot trong môi trường của nó*. Vị trí của robot (cụ thể là tọa độ và hướng) được tính từ những số đo hiện tại của các cảm biến đặt trên nó. Tuy nhiên, các cảm biến này dù hoàn hảo đến đâu, cũng còn những nhược điểm so với loại khác và độ chính xác bị hạn chế vì ảnh hưởng của can nhiễu. Do vậy, kết quả định vị nhận được từ số đo của từng cảm biến riêng rẽ thường bị giới hạn về độ chính xác và tin cậy. Ví dụ, *cảm biến siêu âm* là một thiết bị có giá thành thấp và có lợi thế khi cho được kết quả đo nhanh hơn so với các thiết bị khác. Tuy nhiên sự đo khoảng cách bằng cảm biến này bị một số hạn chế cơ bản, đó là chỉ hữu ích khi lập bản đồ ở môi trường trong nhà (indoor) và độ chính xác của kết quả bị ảnh hưởng bởi các hiện tượng như sự trải rộng chùm tia hoặc sai số do hiện tượng “đọc chéo” của chùm sóng siêu âm nên khó xác định được kích thước của vật [15]. Các *camera thu hình* (video camera) hiện cũng được sử dụng rộng rãi trong các robot di động, tuy nhiên các dữ liệu hình ảnh lại bị ảnh hưởng nhiều vào điều kiện ánh sáng và kết cấu bề mặt của các đối tượng. Hơn nữa, hệ thống thông thường này không thể đo trực tiếp các tham số hình học, chẳng hạn như khoảng cách của các đối tượng. Các *camera thu hình nổi* (stereo camera) có thể khắc phục vấn đề này một phần bằng

cách xác định khoảng cách gián tiếp trên cơ sở tính toán các tọa độ điểm ảnh, nhưng lại đòi hỏi một quá trình tính toán lớn với độ chính xác của kết quả không cao và cũng lệ thuộc nhiều về điều kiện ánh sáng. Thế giới hiện đã bắt đầu sử dụng loại *cảm biến đo xa laser LRF (laser range finder)* dựa trên nguyên tắc xác định thời gian đi-về của một xung laser phản xạ từ vật cản cho phép xác định được khoảng cách đến vật. LRF có lợi thế là nó có khả năng thu thập đo đạc khoảng cách với tốc độ và độ chính xác cao, kết quả không phụ thuộc quá nhiều vào điều kiện môi trường, trừ trường hợp cảm biến này bị “mù” với các vật liệu trong suốt. Tuy nhiên, thông tin hình ảnh 2D do cảm biến laser đem lại có thể không đủ trong một số trường hợp cần phát hiện các vật có kết cấu không giống nhau theo chiều dọc (như bàn, các khung dầm ngang...). Thêm nữa, một loại cảm biến phổ biến và dễ sử dụng, dùng để đo trực tiếp góc quay (hướng) của robot, đó là cảm biến chỉ hướng từ - địa bàn. Tuy nhiên do từ trường của trái đất thường bị bóp méo gần đường tải điện hoặc kết cấu thép [21] làm ảnh hưởng trực tiếp đến phép đo góc khi sử dụng cảm biến này ở môi trường trong nhà, hoặc nơi có nhiều vật thể kim loại,...

Những ưu nhược điểm của các cảm biến được nêu ra như trên là lý do trong những thập niên gần đây phương pháp *tổng hợp cảm biến (sensor fusion)* đã được áp dụng nhằm nâng cao độ chính xác và tin cậy của các ước lượng trạng thái robot. Tổng hợp cảm biến chính là việc kết hợp số liệu ra từ nhiều nguồn dữ liệu khác nhau theo một cách nào đó để tạo nên được một bức tranh về thế giới quanh robot trung thực và mạch lạc hơn. Mục đích của tổng hợp cảm biến là thực hiện một kiến trúc cảm nhận mới trên cơ sở tổng hợp đa thông tin từ cảm biến cho việc nhận dạng môi trường. Ví dụ trong trường hợp robot di động, kết quả ước lượng vị trí robot trong môi trường dựa trên các kỹ thuật tổng hợp cảm biến sẽ được chính xác và tin cậy hơn. Để thực hiện phương pháp tổng hợp cảm biến cho dẫn đường robot, ngoài việc thiết kế phần cứng với các cảm biến hiện đại thì việc phát triển các giải thuật phần mềm tổng hợp số liệu cảm biến là quan trọng. Hiện nay, hầu hết các giải thuật này được phát triển dựa trên suy luận xác suất. Áp dụng bộ lọc Kalman mở rộng [2,3] là giải pháp xác suất hiệu quả nhất để ước tính đồng thời các vị trí của robot

dựa trên một số thông tin về cảm biến bên trong và bên ngoài robot. Một trong những nghiên cứu quan trọng của Luận án này là tập trung cho việc áp dụng phương pháp tổng hợp cảm biến để nâng cao độ chính xác ước lượng vị trí của robot trên quỹ đạo chuyển động. Tiếp đó là việc nghiên cứu áp dụng tiếp kết quả này cho các khâu còn lại trong dẫn đường là việc xây dựng bản đồ, vạch đường đi và điều khiển ổn định robot đi tới đích.

2. Sự phát triển các phương pháp tổng hợp cảm biến cho bài toán dẫn đường robot di động trên thế giới và ở Việt Nam.

Nghiên cứu phương pháp tổng hợp cảm biến dùng cho kỹ thuật dẫn đường robot di động ở trong nhà hoặc ngoài trời đã được một số tác giả quan tâm ngay từ những năm đầu thập niên 90 của thế kỷ trước [11,106]. Với các hoạt động của robot ở môi trường trong nhà hoặc để nhận dạng các vật thể gần, cảm biến siêu âm thường được sử dụng. Để bù trừ cho nhược điểm của các cảm biến siêu âm hầu hết các tác giả bổ sung bằng một cảm biến camera [82,95,130] và [32,84], các phương pháp tổng hợp này thường dùng phương pháp xác suất Bayes kết hợp với bản đồ chia lưới,... Gần đây với sự phát triển của cảm biến laser, ở hệ thống này laser chiếu vào đối tượng tại một góc xác định, để camera chuyên dụng ghi lại góc của ánh sáng phản chiếu. Sau đó, vị trí của các điểm dọc theo đường chiếu sẽ được tính qua phép đặc tam giác [18,23]. Tuy nhiên, phép đặc tam giác laser không thể thực hiện tốt nhiệm vụ với vật liệu bóng và trong suốt (như kính), nhưng kết hợp với các cảm biến dẫn đường thị giác 3D thì sẽ khắc phục được nhược điểm nói trên. Để thu thập được thông tin 3D một số tác giả đã cố gắng đi sâu vào các giải thuật xử lý Lacroix [78], thuật toán đặc tam giác động Perceptron [97] hoặc cải tiến công nghệ từ một máy quét 2D tiêu chuẩn thành máy quét 3D Surmann [108]; Hahnel và Burgard [58]; và [56,125]. Hầu hết các công trình này chỉ chú ý ứng dụng cho lập bản đồ toàn cục để robot tránh vật cản, chưa đi sâu vào định vị chính xác vị trí của robot. Gần đây, một số tác giả đã nghiên cứu áp dụng bộ lọc Kalman mở rộng (EKF) kết hợp phương pháp đo lường *odometric* với các cảm biến: đo xa laser, gyroscope, compass, camera, v.v...để nâng cao độ tin cậy của sự ước lượng trạng thái của robot.

Tuy nhiên các tác giả thường chỉ dừng lại tổng hợp từ hai loại cảm biến với nhau; chưa tổng hợp được nhiều cảm biến vì liên quan đến thời gian xử lý thu thập dữ liệu khác nhau trên mỗi cảm biến. Thí dụ như, Jensfelt [67]; Grossman [52];... các tác giả đã sử dụng giải trọng số biến đổi Hough để phát hiện các đường thẳng là các bức tường như là các điểm mốc khi tích hợp nhiều phép đo xa của dữ liệu laser ở trong nhà. Hay là, Arras và Vestli [19] và Arras và Tomatis [20] chế tạo được robot có khả năng tự động đi dọc hành lang tòa nhà. Trong trường hợp này các bức tường ở phía xa được phát hiện bởi máy đo xa laser trong khi các cạnh thẳng đứng được phát hiện bởi camera ảnh. Một phương pháp khác của Zhou Xiaowei [133] sử dụng các cột mốc nhân tạo bằng cách trích chọn các điểm gãy (breakpoint) từ tập dữ liệu máy đo xa laser và dữ liệu đo odometry [24,104]. Mới đây, Panich và Afzulpurkar [98] đã tích hợp các cảm biến gyroscope, compass với hệ thống CaPS (*Camera Positioning System*) để định vị robot nhưng rất cồng kềnh, phụ thuộc nhiều vào điều kiện môi trường.

Cũng như các tác giả trên thế giới, các tác giả ở Việt Nam cũng mới chỉ dừng lại nghiên cứu tổng hợp từ một vài cảm biến như cảm biến định vị toàn cầu GPS và cảm biến dẫn đường quán tính INS cho dẫn đường các phương tiện cơ giới ngoài trời, tuy nhiên trường hợp này đòi hỏi sai số khoảng cách lớn (một vài chục mét) [1,9,11]. Cũng như vậy với các bài toán dẫn đường trong nhà của một số tác giả [3] đã sử dụng camera lập thể KINECT của Microsoft cung cấp được độ sâu và khoảng cách của đối tượng bởi cảm biến hồng ngoại tích hợp sẵn trong camera để phát hiện nhận dạng các điểm mốc bằng các màu sắc nhân tạo rất phức tạp phụ thuộc nhiều vào điều kiện ánh sáng. Hay như [5], tác giả đã sử dụng kết hợp nhiều cảm biến (như camera, siêu âm, từ địa bàn, GPS, lập mã quang) để nhận dạng môi trường cho người tàn tật điều khiển xe điện nhưng chưa thực sự đi sâu vào bài toán định vị chính xác vị trí của xe. Hay là, nhóm tác giả ĐH Quốc tế TP.HCM đã sử dụng một camera lập thể (stereo) nhận diện các vật cản và một hệ thống máy tính để xử lý hình ảnh, khi gặp vật cản xe sẽ tự động tránh và tự chọn hướng đi phù hợp, hoặc [Ngô Văn Thuyên, Đề tài cấp Bộ GD-ĐT, mã số: B2009-22-37] sử dụng vòng 12

cảm biến siêu âm để định vị và dẫn đường cho robot di động trong nhà, hệ thống phức tạp, nhưng lại chỉ cho kết quả có tính chất định tính là nhiều. Một điểm nữa, hầu như chưa thấy có nhóm tác giả nào ở Việt Nam tập trung nghiên cứu đưa cảm biến đo xa laser LRF và cảm biến ảnh camera toàn phương vào bài toán tổng hợp cảm biến dẫn đường cho robot di động. Ngoài ra với vấn đề lập bản đồ và tránh vật cản, các tác giả cũng chỉ mới dừng lại với các phân vùng ảnh 2D. Một vấn đề rất quan trọng trong bài toán dẫn đường cho robot di động tự trị là điều khiển ổn định cho robot cũng chưa được nhiều các tác giả trong nước quan tâm [116]; trong khi, như Luận án này sẽ trình bày, khi áp dụng thích hợp kỹ thuật tổng hợp cảm biến sẽ có khả năng đảm bảo độ ổn định quỹ đạo chuyển động dùng hàm Lyapunov của robot.

3. Mục đích nghiên cứu của luận án.

Từ các xuất phát điểm nêu trên, Luận án tập trung vào mục đích sau:

Nghiên cứu thực nghiệm, đề xuất và tiến hành một số phương pháp sử dụng kỹ thuật tổng hợp cảm biến để nâng cao độ chính xác và tin cậy của phép định vị, lập bản đồ và điều khiển chuyển động một mô hình robot di động được thiết kế xây dựng tại phòng thí nghiệm.

Một mô hình robot di động đa cảm biến, có 2 bánh xe kiểu vi sai, được điều khiển PID cấp thấp, được thiết kế xây dựng để làm nền tảng cho việc nghiên cứu áp dụng các thuật toán điều khiển phần mềm với phương pháp tổng hợp cảm biến. Trong quá trình phát triển mô hình hệ thống, việc chuyển đổi hệ đo xa 2D với cảm biến laser thành 3D được tác giả tự thực hiện, đem lại hiệu quả cao, giá thành thấp cho hệ đo đa cảm biến.

Sử dụng mô hình robot di động này, các giải thuật định vị dựa trên bản đồ được áp dụng thực nghiệm. Cụ thể, phương pháp tổng hợp đa cảm biến dựa trên bộ lọc Kalman mở rộng đã được tiến hành với các loại cảm biến: lập mã trục quay, cảm biến chỉ hướng từ-địa bàn, cảm biến đo xa laser và cảm biến ảnh toàn phương. Các kết quả đã được kiểm chứng qua mô phỏng và thực nghiệm tại hiện trường.

Một phương pháp xây dựng bản đồ dẫn đường bằng tổng hợp dữ liệu cảm biến trong không gian đo xa 3D cũng đã được đề xuất phát triển và thử nghiệm.

Hơn nữa, các kết quả tổng hợp dữ liệu cảm biến này cũng được sử dụng làm tiền đề cho việc phát triển các thuật toán bám quỹ đạo ổn định tiệm cận theo tiêu chuẩn ổn định Lyapunov, các kết quả được mô phỏng và thực nghiệm trên robot di động hai bánh vi sai tại phòng thí nghiệm.

Những nội dung trên sẽ góp một phần vào việc bổ sung những nghiên cứu mới cho định vị và dẫn đường robot di động dựa trên kỹ thuật tổng hợp cảm biến hiện nay trên thế giới và ở Việt nam.

4. Cấu trúc của Luận án.

Trên cơ sở mục tiêu đặt ra như trên, ngoài *phần mở đầu* là phần giới thiệu và *phần kết luận*, nội dung chính của Luận án được bố cục thành các chương như sau:

Chương 1: Phương pháp tổng hợp cảm biến

Chương 2: Xây dựng mô hình robot di động đa cảm biến

Chương 3: Tổng hợp cảm biến dùng cho định vị và lập bản đồ dẫn đường robot di động

Chương 4: Điều khiển chuyển động

Chương 1 trình bày tổng quan khái niệm và các phương pháp tổng hợp cảm biến dùng cho robot di động. Các khái niệm cơ bản về các phương pháp định vị robot dựa trên suy luận xác suất và bộ lọc Kalman cũng sẽ được trình bày.

Chương 2 trình bày về quá trình thiết kế chế tạo một robot di động được lắp đặt nhiều cảm biến hiện đại. Bộ phận động lực của robot với các mô-tơ điều khiển các bánh xe theo kiểu vi sai, các mạch điện tử điều khiển robot được thiết kế trên cơ sở các chip vi điều khiển sẽ được trình bày. Nhằm phát triển thêm khả năng phát hiện của robot, một bộ phận cơ khí và chương trình điều khiển để chuyển đổi hệ đo xa 2D với cảm biến laser thành hệ đo xa 3D với các kết quả đo kiểm và thảo luận cũng được báo cáo ở phần cuối chương này.

Chương 3 là phần quan trọng của Luận án trình bày về nghiên cứu thực nghiệm phương pháp tổng hợp cảm biến với việc áp dụng bộ lọc Kalman để ước

tính đồng thời tọa độ và hướng của robot, nhằm nâng cao độ chính xác của phép định vị. Tiếp đó, việc xây dựng một thuật toán mới cho phép xây dựng một bản đồ dẫn đường 2D bằng phương pháp tổng hợp các dữ liệu cảm biến 3D theo không gian sẽ được trình bày. Việc kiểm chứng thành công bản đồ này qua áp dụng các phương pháp vạch đường đi như giải thuật A*[56] hay đồ thị Voronoi [28] trên nó sẽ khép lại một việc sử dụng phương pháp tổng hợp cảm biến cho khâu định vị và lập bản đồ.

Chương 4 báo cáo về việc thực thi một giải pháp sử dụng luật điều khiển ổn định tiệm cận theo tiêu chuẩn Lyapunov cho hệ robot di động hai bánh vi sai trong khâu điều khiển chuyển động. Kết hợp với việc thiết kế bộ lọc Kalman xen vào vòng phản hồi cũng cho thấy chất lượng điều khiển được nâng cao.

CHƯƠNG 1

PHƯƠNG PHÁP TỔNG HỢP CẢM BIẾN

Robot dùng số liệu thu được từ các cảm biến để xác định trạng thái của nó trong môi trường, trên cơ sở đó mà ra các quyết định cho các nhiệm vụ dẫn đường như định vị và lập bản đồ. Tuy nhiên, trong một hệ thống chứa nhiều yếu tố ngẫu nhiên chưa có loại cảm biến nào hoàn hảo để thực hiện nhiệm vụ này một cách thật chính xác. Vì vậy kỹ thuật *tổng hợp cảm biến* (sensor fusion) đã được sử dụng, đó là việc sử dụng đồng thời các dữ liệu từ một cảm biến hay từ nhiều cảm biến rồi lấy thông tin hợp nhất ra để tạo nên bức tranh về trạng thái môi trường chính xác hơn. Ngoài ra trong quá trình điều khiển chuyển động robot ổn định bám theo quỹ đạo, việc áp dụng kỹ thuật *tổng hợp cảm biến* trong trường hợp này cũng cho phép loại bỏ bớt can nhiễu, tăng hiệu quả về điều khiển ổn định [116].

Có nhiều vấn đề phát sinh khi giải quyết vấn đề tổng hợp cảm biến như độ bất định vốn có trên các phép đo của mỗi cảm biến, tính đa dạng về thời gian và không gian của mỗi phép đo. Độ bất định của các số liệu trong các cảm biến không chỉ phát sinh từ tính không chính xác và nhiễu trong các phép đo, mà nó cũng bị gây ra từ sự không rõ ràng và không đồng nhất của môi trường, và không có khả năng phân biệt giữa chúng. Các biện pháp được sử dụng để tổng hợp dữ liệu từ các cảm biến sẽ loại bỏ độ bất định trên, đưa vào tính toán các thông số môi trường ảnh hưởng đến các phép đo cảm biến và kết hợp tính tự nhiên khác nhau của thông tin để có được một thông tin chính xác mô tả môi trường phù hợp nhất. Đó là lý do các phương pháp tổng hợp cảm biến đã được thực hiện trong robot hiện đại để tăng độ chính xác của đo lường, nâng cao độ tin cậy của các ước lượng định vị robot. Các thuật toán tổng hợp cảm biến thường được phân loại thành ba nhóm [53]:

- *Suy luận xác suất*, thường dựa trên: Lý thuyết suy luận Bayesian và lý thuyết Dempster-Shafer; Lý thuyết thống kê; Lý thuyết vận hành đệ qui.
- *Bình phương tối thiểu*, là các phương pháp dựa trên: Bộ lọc Kalman; Lý thuyết tối ưu...

- *Tổng hợp thông minh*, là các phương pháp dựa trên: Logic mờ; Mạng neuron; Các thuật toán di truyền.

Các tác giả cố gắng mô hình hóa độ bất định trong các phép đo trên mỗi cảm biến. Với nhóm *suy luận xác suất*, thường sử dụng phương pháp suy luận Bayesian với các xác suất chiếm giữ của các vật cần được gán lên các ô lưới của bản đồ chia lưới [7,82]. Như vậy đòi hỏi máy tính phải có bộ nhớ đủ lớn và khả năng tính toán cao. Tuy nhiên, độ bất định của phép đo không loại bỏ được hết do xác suất chiếm giữ các ô ở vùng đường bao khó xác định. Cùng tồn tại với lý thuyết Bayes là lý thuyết Dempster-Shafer [8], đây là lý thuyết cho phép giải quyết kỹ hơn về sự kiện không chắc chắn sắp xảy ra. Tuy nhiên trong phương pháp này các phần tử tính toán sẽ tăng lên cấp lũy thừa theo số cảm biến trong hệ thống và như vậy rất khó tính toán.

Trong một số bài toán khi mà dữ liệu đầu vào bị nhiễu, đòi hỏi phải có một phương pháp có khả năng đưa ra quyết định dựa trên những điều kiện không chắc chắn, tức là phải mở rộng từ việc đánh giá định lượng giá trị vật lý đến việc đánh giá theo xác suất hiện trên kết quả tổng hợp của nhiều dữ liệu cảm biến trong không gian một và nhiều chiều. Ở đây áp dụng kỹ thuật *logic-mờ* là một phương pháp hữu ích [9]. Tuy nhiên, các cảm biến không giống nhau, nhiều công việc cần phải thực hiện để thực hiện suy diễn trong bài toán tổng hợp dữ liệu từ các nguồn khác nhau, đòi hỏi hệ thống có khả năng tự tạo ra các qui tắc riêng để tổng hợp dữ liệu. Khả năng học hỏi và thích nghi của mạng neuron cho phép có thể sử dụng được cho mục đích này. Như vậy, muốn giải quyết những yêu cầu đặt ra cần phải có hệ suy diễn neuron - mờ thích nghi, bài toán lại trở về sự phụ thuộc số lượng cảm biến đầu vào. Số lượng cảm biến tăng thì tăng độ tính toán và rất phức tạp; đôi khi không tính được vì phụ thuộc đồng thời các cảm biến.

Một phương pháp hiệu quả để ước tính đồng thời các giá trị đo của các cảm biến đó là bộ lọc Kalman mở rộng (EKF). Bộ lọc này với các phép tính ma trận có thể giảm được khả năng tính toán so với các hàm mũ ở các phương pháp nêu trên.

Sự phụ thuộc không giống nhau của mỗi cảm biến cũng được giải quyết bằng cách tính toán thời gian xử lý khác nhau (độ trễ) trên mỗi cảm biến.

Nội dung chương này sẽ trình bày các khái niệm liên quan đến vấn đề tổng hợp dữ liệu cảm biến. Các phương pháp tổng hợp cảm biến cho robot di động như phương pháp lập bản đồ, định vị robot bằng phương pháp dựa trên lý thuyết *suy luận xác suất Bayesian* và phương pháp EKF để nâng cao phép định vị và hiệu quả điều khiển robot cũng sẽ được trình bày.

1.1. Hệ thống tổng hợp dữ liệu cảm biến.

1.1.1. Tổng hợp dữ liệu đa cảm biến.

Tổng hợp dữ liệu đa cảm biến (Multisensor Data Fusion) là quá trình tổng hợp dữ liệu và thông tin từ nhiều *nguồn dữ liệu* cảm biến khác nhau, nhằm mục đích đưa ra được đánh giá tốt nhất về đại lượng đang xem xét [37-41][61-62][128].

Nguồn dữ liệu là nguồn thu thập được từ các cảm biến và các cơ sở dữ liệu đã có (có thể là cơ sở dữ liệu trên máy tính hoặc văn bản), hoặc thu thập qua nhiều lần lấy dữ liệu khác nhau trên cùng một cảm biến (như các lần thu thập dữ liệu tiên nghiệm và thu thập dữ liệu hiện tại). Ngày nay, các ứng dụng tổng hợp dữ liệu đa cảm biến đã trở nên phổ biến. Các ứng dụng trong dịch vụ thương mại như: điều khiển giao thông, điều khiển robot công nghiệp, robot di động, phân tích tài chính, dự đoán thời tiết, các ứng dụng trong y tế như chuẩn đoán bệnh..., cũng như các ứng dụng trong quân sự: tự động nhận dạng mục tiêu trong các vũ khí thông minh, dẫn đường xe tự hành, giám sát chiến trường...

Các cụm thuật ngữ “tổng hợp cảm biến” (*sensor fusion*); “tổng hợp dữ liệu” (*data fusion*); “tổng hợp thông tin” (*information fusion*); “tích hợp đa cảm biến” (*multisensor integration*) đã được sử dụng rộng rãi trong nhiều tài liệu kỹ thuật, đôi lúc được hiểu nhầm lẫn nhau. Đầu tiên Wald L. [127] đề xuất cụm từ “tổng hợp dữ liệu” thành một thuật ngữ chung, sau đó Bộ Quốc phòng Mỹ [109], và một số công trình của một số tác giả [38,110] cho rằng cụm từ “tổng hợp dữ liệu” biểu thị cho sự tổng hợp dữ liệu thô. Bên cạnh đó, có rất nhiều cuốn sách kinh điển như “Tổng hợp dữ

liệu đa cảm biến” bởi Waltz [128]; Hall David L., Llinas James [61]; Hall David L., McMullen SonyaAH [62]; Mitchell H.B. [85] và Milisavljevic Nada [86]; đề xuất một điều kiện mở rộng, “*tổng hợp dữ liệu đa cảm biến*” và trong cả các cuốn sách đó, thuật ngữ “*tổng hợp dữ liệu*” cũng được xem như “*tổng hợp dữ liệu đa cảm biến*” [62].

Ngoài ra, Dasarathy quyết định sử dụng thêm thuật ngữ “*tổng hợp thông tin*” như thuật ngữ tổng thể cho tổng hợp các loại dữ liệu [39]. Nhưng thuật ngữ *tổng hợp thông tin* đã không được sử dụng rộng rãi, tuy nhiên nó cũng có thể áp dụng trong bối cảnh khai thác dữ liệu và tích hợp cơ sở dữ liệu. Ta có thể định nghĩa thuật ngữ *tổng hợp thông tin* như sau [47]:

Tổng hợp thông tin bao gồm lý thuyết, và các công cụ hình thành, được sử dụng để khai thác sự đồng vận của thông tin từ nhiều nguồn (cảm biến, cơ sở dữ liệu, thông tin thu thập bởi con người, ...) để đưa ra quyết định hay hành động với một ý nghĩa tốt hơn về mặt chất lượng hoặc số lượng, về độ chính xác, chắc chắn,...so với khi sử dụng một nguồn đơn không khai thác sự đồng vận.

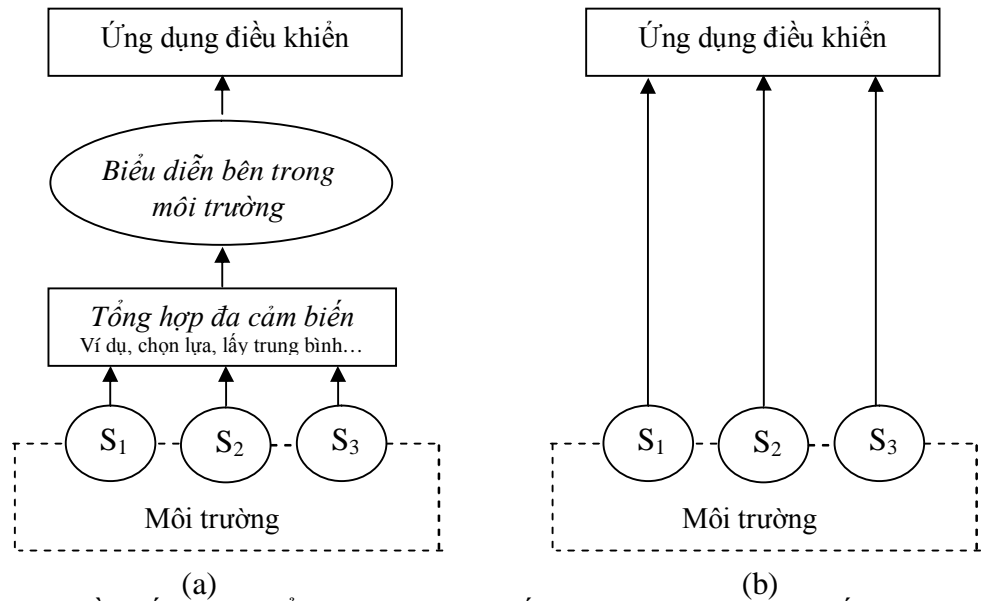
Bằng cách xác định như một tập hợp con của *tổng hợp thông tin*, thuật ngữ *tổng hợp cảm biến* có thể được định nghĩa như sau:

Tổng hợp cảm biến là kết hợp các dữ liệu giác quan hoặc các dữ liệu nhận được để có được kết quả thông tin với một ý nghĩa tốt hơn so với khi sử dụng các nguồn này riêng lẻ.

Ngoài cách định nghĩa trên, McKee [47] phân loại ra hai vấn đề trên như là *tổng hợp trực tiếp*, và *tổng hợp gián tiếp*. Tổng hợp trực tiếp có nghĩa là tổng hợp các dữ liệu cảm biến từ một tập hợp các quá trình thu thập giá trị dữ liệu cảm biến của các cảm biến đồng nhất hoặc không đồng nhất. Trong khi đó *tổng hợp gián tiếp* sử dụng nguồn thông tin như là một kiến thức tiên nghiệm về môi trường và đầu vào của con người. Do đó, *tổng hợp đa cảm biến* mô tả các hệ thống tổng hợp trực tiếp, còn *tổng hợp thông tin* bao gồm các qui trình tổng hợp gián tiếp.

Một thuật ngữ khác thường được sử dụng là *tích hợp đa cảm biến*. Tích hợp đa cảm biến có nghĩa là sự đồng vận sử dụng dữ liệu cảm biến cho sự hoàn thành

nhệm vụ của một hệ thống. Như vậy, sự khác nhau giữa *tổng hợp đa cảm biến* và *tích hợp đa cảm biến* có thể được biểu diễn ở hình 1.1. Trong đó các cảm biến vật lý S_1, S_2, S_3 cung cấp một giao diện để nhận biết môi trường.

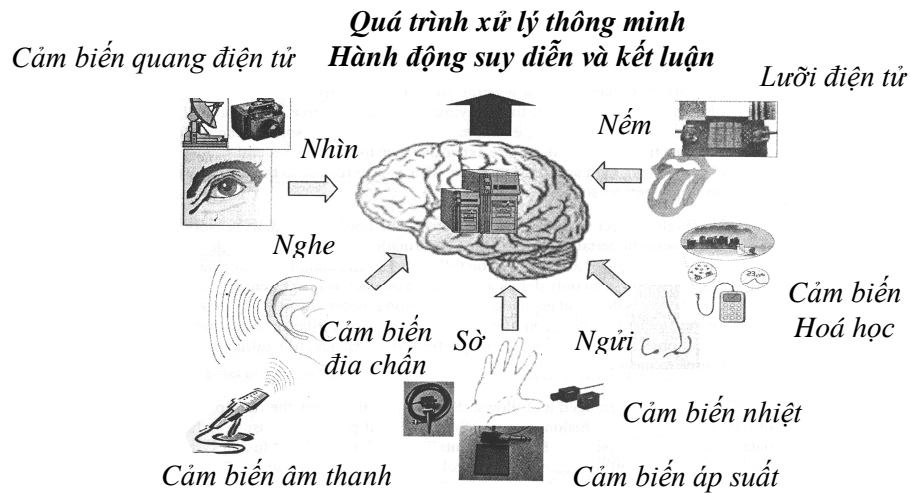


Hình 1.1. Sơ đồ khối của: a) tổng hợp đa cảm biến; b) tích hợp đa cảm biến [47].

Sơ đồ 1.1(a) biểu diễn dữ liệu cảm biến được chuyển đổi bởi khối *tổng hợp đa cảm biến* thành một đại lượng tương ứng của các biến số của quá trình nhận biết môi trường. Dữ liệu này sau đó được sử dụng cho ứng dụng điều khiển. Ngược lại, sơ đồ 1.1(b) minh họa ý nghĩa của *tích hợp đa cảm biến*, tại đó các dữ liệu cảm biến khác nhau điều khiển trực tiếp ứng dụng điều khiển.

Từ những phân tích ở trên, chúng ta có thể nhận xét rằng: khái niệm *tổng hợp dữ liệu đa cảm biến* là một khái niệm mới được đưa ra trong những năm gần đây, tuy nhiên trong lịch sử tồn tại và phát triển của thế giới tự nhiên, tổng hợp dữ liệu đã là một quá trình tự nhiên. Con người và các loài vật có khả năng sử dụng nhiều giác quan để tăng khả năng tồn tại. Quá trình tổng hợp dữ liệu từ nhiều nguồn khác nhau của con người là một quá trình xử lý thông minh. Quá trình này mô tả bằng cách nào bộ não người tổng hợp dữ liệu từ các giác quan của cơ thể ví dụ như mắt, mũi, tay, chân và da. Hệ thống cảm biến của cơ thể thu nhận các dữ liệu từ bên ngoài và qua hệ thần kinh đưa đến bộ não. Bộ não tổng hợp các dữ liệu này để rút ra và hoàn thiện các tri thức về môi trường xung quanh, đưa ra các kết luận hoặc suy

diễn từ việc tổng hợp dữ liệu này [61,92]. Hình 1.2 chỉ ra hệ thống tổng hợp dữ liệu thông minh của bộ não người sử dụng các nguồn thông tin đa cảm biến từ các giác quan [92].



Hình 1.2 Quá trình xử lý thông minh của con người. [92]

Về nguyên tắc, một hệ thống sử dụng nhiều nguồn dữ liệu cảm biến sẽ đưa ra được một kết quả có ý nghĩa hơn là từ một nguồn dữ liệu đơn, vì hệ thống cung cấp đa dạng thông tin bổ sung cũng như thông tin dự phòng.

Ta xét một ví dụ đơn giản một robot chuyển động cần định vị chính xác vị trí của mình nếu chỉ sử dụng *camera lập thể* có thể xác định chính xác góc quay bằng cách ghi lại góc của ánh sáng phản chiếu so với một đối tượng (ví dụ như cột mốc) nhưng không thể đo chính xác khoảng cách đến đối tượng, nhưng máy đo xa laser thì có thể đo được chính xác khoảng cách robot đến đối tượng. Như vậy, nếu kết hợp nguồn dữ liệu nhận được của hai cảm biến trên ta sẽ xác định vị trí và hướng chính xác của robot. Một ví dụ tiếp theo là phương pháp định vị đơn giản thường được sử dụng, đó là phương pháp *odometry*. Phương pháp này gặp phải vấn đề sai số tích lũy [22,50], trong đó sự bất định của vị trí ước lượng bởi hệ thống odometric tăng theo thời gian trong khi robot di chuyển....Tuy nhiên nếu áp dụng phương pháp tổng hợp cảm biến bằng bộ lọc Kalman mở rộng sẽ cải thiện đáng kể độ chính xác ở trên bằng cách tổng hợp dữ liệu với các dữ liệu tiên nghiệm và dữ liệu hiện tại của nó.

Một số lý do chỉ ra tại sao phải tổng hợp dữ liệu cảm biến được nêu ra tóm tắt dưới đây [71]:

- Nâng cao tính bền vững của hệ thống: Trong tình huống nguồn dữ liệu đơn đó hỏng hoàn toàn thì hoạt động của hệ thống sẽ bị ảnh hưởng. Vì vậy, một hệ thống sử dụng một vài nguồn dữ liệu sẽ bền vững hơn trong vận hành hoặc có một tỷ lệ lỗi vận hành thấp hơn các hệ thống chỉ sử dụng một nguồn dữ liệu.

- Tốt hơn trong việc đánh giá tình huống và ra quyết định, và chính vì vậy phản ứng của hệ thống sẽ chính xác hơn. Từ các nguồn dữ liệu khác nhau quá trình tổng hợp dữ liệu sẽ đem lại một đánh giá tốt hơn về tình huống đang xem xét, vì vậy tăng độ chính xác khi kết luận dẫn đến dễ dàng đưa ra quyết định.

- Tăng độ chính xác của dữ liệu, giảm các dữ liệu không chắc chắn và mơ hồ.

- Mở rộng thông tin về đối tượng. Nhiều nguồn dữ liệu sẽ cung cấp thêm thông tin về đối tượng hoặc sự kiện quan sát. Mở rộng thông tin bao gồm cả không gian và thời gian. Ví dụ, thông thường mỗi cảm biến độc lập chỉ quan sát được một vùng không gian và thời gian nhất định, tăng thêm cảm biến tăng thêm tầm phủ quan sát về không gian và thời gian.

- Khắc phục trường hợp khi không có nguồn dữ liệu đơn hoàn chỉnh. Hầu hết các nguồn thu thập dữ liệu, ví dụ như các cảm biến, chỉ thực sự hữu ích trong một môi trường nhất định. Những thông tin thu nhận được từ một nguồn dữ liệu đơn sẽ bị hạn chế và có thể không hoàn toàn đầy đủ và đáng tin cậy. Ví dụ như cảm biến siêu âm chỉ phát hiện được đối tượng khi có tín hiệu âm thanh được phát ra, cảm biến quang điện tử (camera) hoàn toàn phụ thuộc vào điều kiện ánh sáng ở môi trường xung quanh và các cảm biến từ địa bàn phụ thuộc nhiều nhiều từ tính ở môi trường xung quanh...;

- Tăng khả năng chống nhiễu bằng cách tăng số chiều của không gian đo lường. Ví dụ, phép đo có chất lượng mong muốn với các cảm biến quang và cảm biến siêu âm.

- Hiệu quả về chi phí với việc giảm về chi phí tính toán, truyền thông và tài nguyên mạng. Trong trường hợp tổng quát, hầu hết chi phí sẽ phụ thuộc vào nhiều nguồn dữ liệu đầu vào hơn là dựa vào một nguồn dữ liệu để cung cấp tất cả thông tin cần thiết. Ví dụ, để xây dựng một cảm biến thực hiện nhiều chức năng sẽ đắt hơn rất nhiều so với việc kết hợp một vài cảm biến đơn giản và rẻ tiền với một chức năng cụ thể.

- Cải thiện độ phân giải khi nhiều phép đo độc lập của cùng một tính năng được tổng hợp, kết quả giá trị độ phân giải tốt hơn so với duy nhất của một phép đo của một cảm biến.

- Khả năng thu nhỏ sự phức tạp của hệ thống: Trong hệ thống thiết kế truyền thống các phép đo cảm biến thường được đưa trực tiếp vào các ứng dụng điều khiển, do đó sẽ phải đối phó với một số lượng lớn sự không chính xác, tính nhập nhằng, và luồng dữ liệu không đầy đủ. Trong khi với một hệ thống hiện đại dữ liệu cảm biến được xử lý trước bằng các phương pháp tổng hợp, đầu vào cho ứng dụng điều khiển có thể được chuẩn hóa độc lập với các cảm biến được sử dụng, do đó tạo điều kiện thuận lợi thực hiện ứng dụng và cung cấp khả năng của sự điều chỉnh trong hệ thống cảm biến về số lượng và các loại cảm biến được sử dụng mà không cần sự điều chỉnh của phần mềm ứng dụng [48].

1.1.2. Quản lý đa cảm biến.

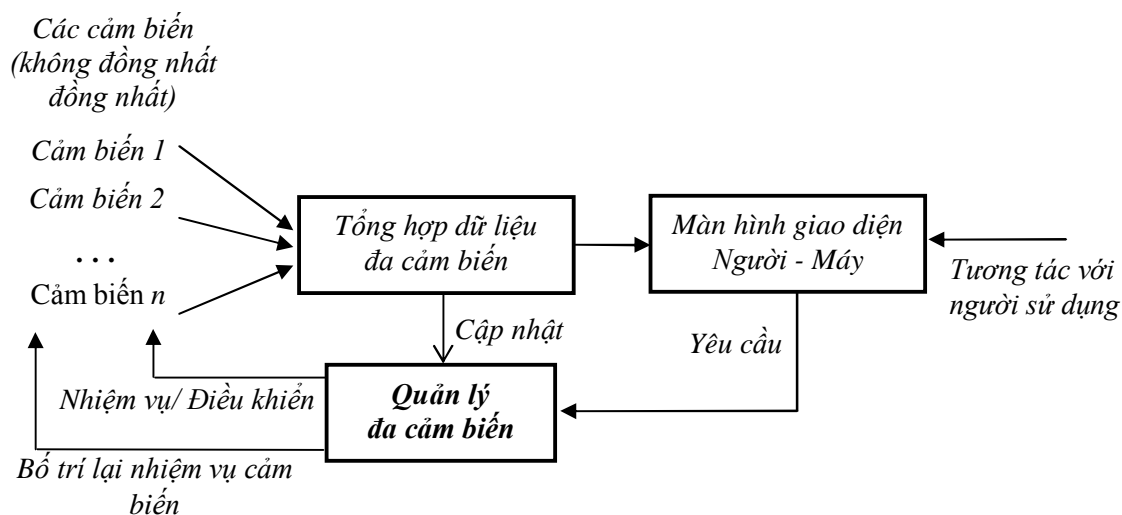
Các cảm biến cung cấp dữ liệu và thông tin cho quá trình tổng hợp dữ liệu. Để đưa ra đánh giá tốt nhất về đại lượng đang xem xét, một trong các yêu cầu đầu tiên là phải lựa chọn được các loại cảm biến phù hợp và quản lý chúng. Quản lý cảm biến giải quyết các công việc phức tạp trong các trường hợp sau [40,63,93]:

- Nguồn dữ liệu từ một cảm biến là không đủ.
- Khả năng của các cảm biến là hạn chế.
- Tính động học của môi trường cao.
- Khả năng và sự thực thi của các cảm biến là khác nhau.
- Khả năng xử lý tính toán hạn chế.
- Lỗi trong việc lập lịch các cảm biến.

- Các yêu cầu tổng hợp dữ liệu và thông tin.

Ví dụ như trên nền chuyển động của robot phải có một bộ điều khiển quản lý các cảm biến cho phép lựa chọn, sắp xếp quyền ưu tiên, chỉ định, phân bổ và sắp xếp các nhiệm vụ cho các cảm biến. Mục đích của một hệ thống quản lý đa cảm biến là quản lý, bố trí và tích hợp các cảm biến thông thường lại với nhau nhằm giải quyết các bài toán đặc thù nhất là trong bài toán giám sát đối tượng chuyển động. Quản lý là nhằm điều khiển các cảm biến; bố trí mang lại hiệu quả nhất khi sử dụng các cảm biến; và tổng hợp mang lại khả năng phối hợp giữa các cảm biến trong hệ thống hoặc là kết hợp toàn bộ các cảm biến lại với nhau. Nghĩa là hệ thống quản lý đa cảm biến phải lựa chọn các cảm biến để thực hiện đúng được công việc cho từng đối tượng cụ thể trong một thời gian chính xác nhằm tối ưu hoá tất cả các công việc của hệ thống tổng hợp dữ liệu. Nói một cách khác, nó cần giải quyết các vấn đề dưới đây:

- Sử dụng loại cảm biến nào hoặc nhóm cảm biến nào ?
- Hoạt động theo chế độ nào (mode) hoặc nhiệm vụ nào ?
- Làm thế nào để điều khiển được cảm biến ?
- Khi nào thì bắt đầu ?



Hình 1.3 Hệ thống quản lý đa cảm biến.

Quản lý đa cảm biến được thiết kế từ việc cấp phát các tài nguyên hoặc đặt lịch cho các nhiệm vụ của hệ thống. Hình 1.3 mô tả một sơ đồ khối đặc trưng quản

lý đa cảm biến. Với các tiên bộ hiện nay trong công nghệ chế tạo cảm biến, vai trò và chức năng của chúng đã được mở rộng. Chúng ta có thể phân loại vai trò và chức năng của cảm biến theo tính hệ thống và theo cấu trúc định trước.

1.1.3. Phân loại tổng hợp đa cảm biến.

1.1.3.1. Phân loại theo mức.

Trong quá trình tổng hợp dữ liệu và thông tin, quản lý các cảm biến có thể phân loại theo 03 mức dưới đây [71]:

- *Mức thấp*: mức thấp nhất của quản lý đa cảm biến liên quan đến việc điều khiển riêng lẻ từng cảm biến một, ví dụ như hướng theo dõi, điểm quan sát, thay đổi tần suất theo dõi... Ví dụ, một radar định hướng theo dõi một mục tiêu có thể được dẫn hướng bởi một hệ thống quản lý cảm biến. Hay nói cách khác, tổng hợp nhiều nguồn dữ liệu thô để tạo ra một nguồn dữ liệu mới mà kỳ vọng mang nhiều thông tin hơn các thông tin đầu vào.

- *Mức trung*: tại mức trung hệ thống quản lý đa cảm biến sẽ tập trung vào nhiệm vụ và các mode hoạt động của các cảm biến. Các cảm biến sẽ hoạt động theo phương thức ưu tiên các nhiệm vụ và xác định khi nào và bằng cách nào một mode có thể được kích hoạt. Bên cạnh đó, hệ thống quản lý các cảm biến có thể xem xét một vài chức năng cơ bản của cảm biến để thay đổi cảm biến khi thực hiện một chức năng nào đó, (ví dụ, tổng hợp các tính năng khác nhau như: các cạnh, các góc, các đường thẳng, các kết cấu, hoặc các vị trí vào một bản đồ tính năng...). Mức trung phải giải quyết các nhiệm vụ sau:

+ Sắp lịch cho các nhiệm vụ của cảm biến.

+ Điều khiển cảm biến, các hỗ trợ để thu thập thêm dữ liệu từ các cảm biến khác.

+ Thay đổi chế độ làm việc của cảm biến.

- *Mức cao*: Mỗi một cảm biến chỉ có khả năng tìm kiếm một lượng thông tin hạn chế. Tuy nhiên, hệ thống quản lý đa cảm biến có thể trực tiếp điều khiển một vài cảm biến trong một mô hình tích hợp hoặc tổng hợp dữ liệu từ các cảm biến

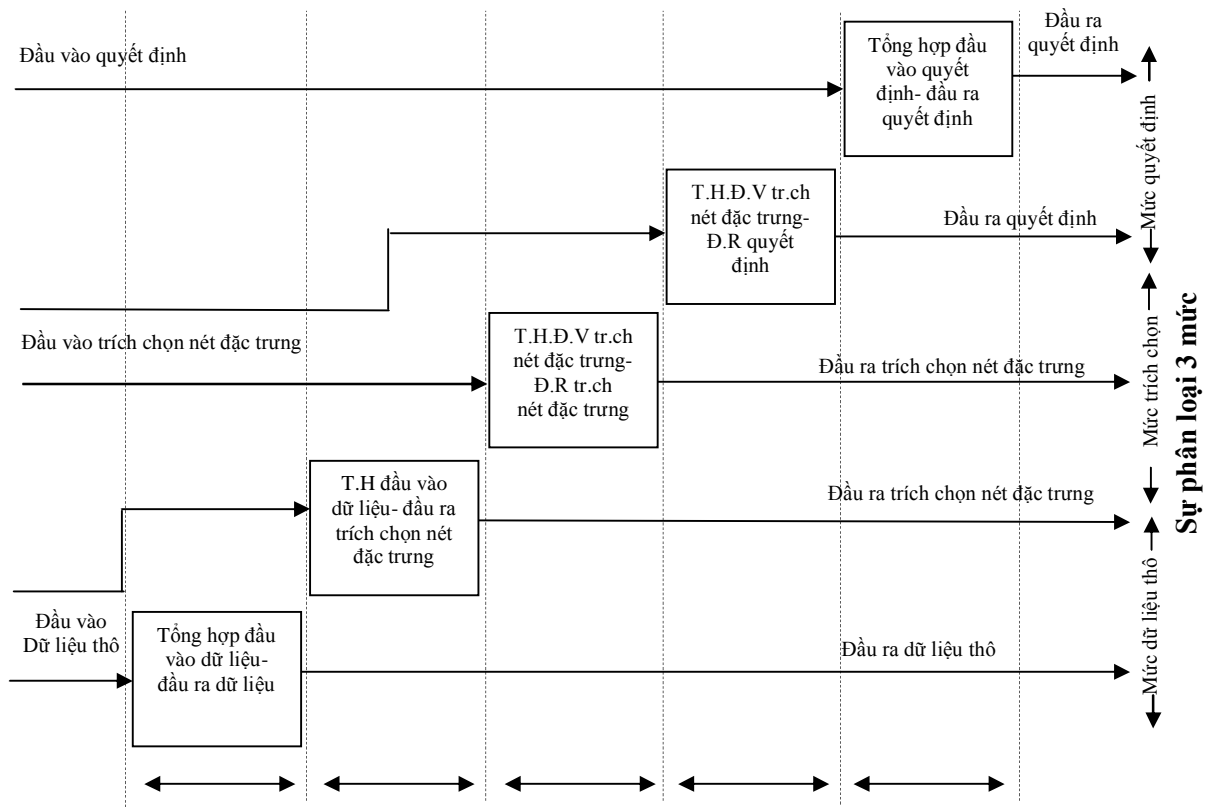
thành một nguồn cung cấp thông tin cho quá trình tổng hợp. Nhiệm vụ của mức này là:

- + Sắp xếp hoặc triển khai vị trí của các cảm biến động sao cho vùng bao phủ là tốt nhất.

- + Tổng hợp dữ liệu của các cảm biến một cách hiệu quả và tối ưu (cực tiểu hoá các bố trí không cần thiết). Tổng hợp các quyết định từ một vài hệ chuyên gia, các phương pháp đó bao gồm: chọn lựa, logic-mờ và các phương pháp thống kê.

1.1.3.2. Phân loại dựa trên dữ liệu vào ra.

Dasarathy đề xuất sự phân loại được chọn lọc dựa trên mô hình 3 mức [38]. Đó là phân loại quá trình tổng hợp suy ra từ mức độ trừu tượng khác nhau của dữ liệu đầu vào và đầu ra. Ví dụ, việc xử lý là sự lựa chọn và trích chọn nét đặc trưng từ các mức dữ liệu thô do vậy các kết quả thuộc về mức độ nét đặc trưng; Hay là, sự nhận dạng và quá trình nhận dạng hoạt động giữa mức độ nét đặc trưng và mức độ quyết định. Những mô hình tổng hợp mơ hồ này đôi khi được gán theo mức của dữ liệu đầu vào và đôi khi theo mức của dữ liệu đầu ra. Để tránh những vấn đề phân loại này, Dasarathy mở rộng tổng quan ba mức thành năm loại tổng hợp được xác định bởi các đặc điểm đầu vào và đầu ra. Hình 1.4 mô tả mối quan hệ giữa việc phân loại 3 cấp với mô hình Dasarathy.



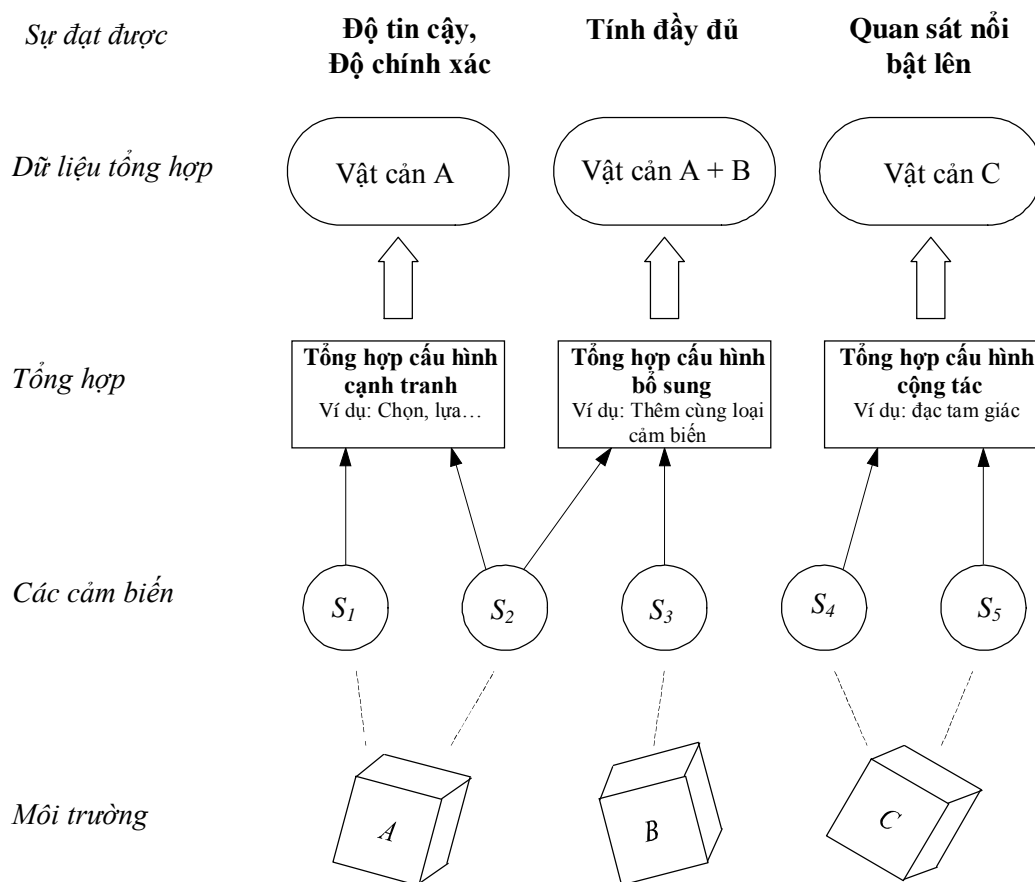
Sự phân loại bởi Dasarathy

Hình 1.4 Các đặc điểm tổng hợp khác nhau dựa trên đầu vào /ra [38].

1.1.3.3. Phân loại dựa trên cấu hình cảm biến.

Hệ thống tổng hợp cảm biến cũng có thể được phân loại theo các kiểu cấu hình của cảm biến. Durant-Whyte phân biệt ba loại cấu hình cảm biến [41]:

Cấu hình bổ sung: Một cấu hình cảm biến được gọi là cấu hình *bổ sung* nếu các cảm biến không phụ thuộc trực tiếp vào nhau, nhưng có thể được kết hợp để cung cấp một hình ảnh đầy đủ hơn về hiện tượng quan sát. Điều này giải quyết sự không đầy đủ của dữ liệu cảm biến. Một ví dụ cho cấu hình bổ sung đó là việc sử dụng nhiều camera để quan sát từng phần của một phòng như ứng dụng trong [64]. Nói chung, tổng hợp dữ liệu cấu hình bổ sung là dễ dàng, do dữ liệu từ các cảm biến độc lập có thể được cộng thêm với nhau [26]. Cảm biến S_2 và S_3 trong hình 1.5 cấu hình bổ sung, do đó mỗi cảm biến quan sát mỗi phần khác nhau của không gian môi trường.



Hình 1.5 Tổng hợp cấu hình: cạnh tranh, bổ sung và cộng tác. [47]

Cấu hình cạnh tranh: Các cảm biến là cấu hình cạnh tranh nếu mỗi bộ cảm biến cung cấp các phép đo độc lập của cùng một nét đặc trưng của vật cân. Visser và Groen [123] phân biệt ra hai cấu hình cạnh tranh - sự tổng hợp của dữ liệu từ các bộ cảm biến khác nhau hoặc tổng hợp của các phép đo từ một cảm biến đơn nhận được tại các thời điểm khác nhau. Cấu hình cảm biến cạnh tranh còn được gọi là *cấu hình dư* [79].

Một trường hợp đặc biệt của tổng hợp cảm biến cạnh tranh là *dung sai* (sai hỏng cho phép). Dung sai đòi hỏi một đặc tính chính xác của sự vận hành và các mô hình sai số của hệ thống. Trong trường hợp của một sai số được bao phủ bởi giả thuyết sai số, hệ thống vẫn còn để cung cấp sự vận hành đã xác lập của nó. Ví dụ cho cấu hình dung sai là dự phòng N mô-dun [94] và các hệ thống khác mà một số lượng nhất định của các thành phần sai số được chấp nhận [87]. Các cảm biến S_1 và

S_2 trong hình 1.5 biểu diễn một cấu hình cạnh tranh, trong đó cả hai cảm biến dự phòng quan sát cùng một nét đặc trưng của một đối tượng trong không gian môi trường.

Cấu hình cộng tác: Một hệ thống cảm biến cộng tác là sử dụng thông tin được cung cấp bởi hai cảm biến độc lập để dẫn xuất ra một thông tin mà không có sẵn từ các cảm biến đơn. Một ví dụ cho cấu hình cộng tác là bằng kết hợp hình ảnh 2D từ hai camera để suy ra được một hình ảnh 3D. Theo Brooks và Iyengar [26] tổng hợp cảm biến cộng tác là rất khó để thiết kế, bởi vì kết quả dữ liệu là rất nhạy cho sự không chính xác trong tất cả cảm biến tham gia độc lập. Như vậy, trái ngược với tổng hợp cấu hình cạnh tranh, tổng hợp cấu hình cộng tác thường làm giảm độ chính xác và độ tin cậy.

Các cảm biến S_4 và S_5 trong hình 1.5 biểu diễn cho một cấu hình cộng tác. Cả hai cảm biến quan sát cùng một đối tượng, nhưng các phép đo được sử dụng để tạo một quan sát nổi bật lên trên đối tượng C mà không thể suy ra từ các phép đo đơn S_4 và S_5 .

Ba loại của cấu hình cảm biến này không loại trừ lẫn nhau. Càng thêm nhiều ứng dụng sẽ thêm nhiều khía cạnh hơn so với chỉ một ứng dụng trong ba loại cấu hình. Một ví dụ giống như kiến trúc tổng hợp áp dụng nhiều camera để giám sát một khu vực nhất định. Khu vực được bao phủ bởi hai hoặc nhiều cảm biến camera cấu hình cảm biến lúc này có thể là cấu hình cạnh tranh hoặc cấu hình cộng tác. Nhưng đối với vùng chỉ quan sát với một cảm biến camera thì là cấu hình bổ sung.

1.1.4. Các vấn đề cần giải quyết trong bài toán tổng hợp dữ liệu đa cảm biến.

Tổng hợp dữ liệu phải xử lý nhiều nguồn thông tin khác nhau, các thông tin có thể có nhiều, độ chính xác thấp, số lượng lớn... Dưới đây là các vấn đề thông thường cần phải giải quyết khi tổng hợp dữ liệu [40], [61]:

- Số chiều và sự sắp xếp của dữ liệu. Các bộ cảm biến khác nhau có dữ liệu được đo lường khác nhau và vì vậy có số chiều và các đặc trưng khác nhau. Việc sắp xếp và biến đổi dữ liệu là rất cần thiết để có được định dạng và chuẩn chung.

Quá trình đăng ký và nhập dữ liệu được xem là một phần của quá trình sắp xếp dữ liệu. Các hệ thống cảm biến khác nhau sẽ đưa ra các dữ liệu khác nhau trong các mặt sau:

- + Hệ thống tọa độ: dữ liệu có thể thu được ở các hệ tọa độ khác nhau ví dụ như trong các hệ tọa độ gắn liền, hệ tọa độ quán tính, hệ tọa độ dẫn đường...
- + Đơn vị đo: mét, độ góc,...vv...
- + Kích thước khác nhau của dữ liệu.
- + Đặc trưng: tần số, biên độ, dữ liệu ảnh hoặc không phải là dữ liệu ảnh.
- + Độ phân giải và độ chính xác của dữ liệu.
- + Nền tham chiếu: các cảm biến trong cùng một nền hoặc trên các nền khác nhau cần phải được sắp xếp vào trong một không gian tham chiếu chung.

Vì vậy, các kỹ thuật biến đổi chuẩn hoá dữ liệu thành các định dạng và chuẩn chung phải được sử dụng cho các quá trình tổng hợp dữ liệu. Việc sắp xếp dữ liệu là để biến đổi các dữ liệu quan sát được từ nhiều nguồn thành một định dạng chung.

- Bản chất và độ tin cậy của dữ liệu. Việc tổng hợp dữ liệu phân tán và tập trung cần phải xử lý vấn đề về bản chất và độ tin cậy của dữ liệu. Một cách lý tưởng, mỗi dữ liệu được tổng hợp cần phải chứa thông tin, dữ liệu này sẽ dùng cho cái gì và độ tin cậy như thế nào.

- Sắp xếp theo thời gian. Việc đồng bộ hoá thời gian dữ liệu rất quan trọng. Vì chỉ các dữ liệu gần nhau về mặt thời gian mới có thể được tích hợp lại để có kết quả có ý nghĩa. Điều này đặc biệt quan trọng với các dữ liệu xung khắc nhau. Các dữ liệu được tổng hợp phải đảm bảo yêu cầu nhất định về mặt thời gian, yêu cầu này phụ thuộc vào dạng của hệ thống tổng hợp. Các hệ thống tổng hợp theo thời gian thực như hệ thống tổng hợp trên máy bay thì độ gần về mặt thời gian điển hình được tính theo thời gian micro giây hoặc miligiây, trong khi đó các hệ thống tổng hợp không yêu cầu thời gian thực có thể có độ gần thời gian tính theo phút hoặc giờ. Tuy nhiên, đối với các dữ liệu bổ sung thì độ gần về mặt thời gian có thể được yêu cầu thấp hơn do dữ liệu bổ sung vẫn có thể cung cấp thông tin hữu ích cho hệ thống tổng hợp ngay cả khi nguồn dữ liệu không đủ gần về mặt thời gian.

Sắp xếp theo thời gian phức tạp do:

- + Các cảm biến được bố trí ở các vùng địa lý khác nhau hoặc trên các nền khác nhau.

- + Độ chênh lệch thời gian của sự kiện do tốc độ truyền tín hiệu khác nhau, chẳng hạn tốc độ tín hiệu âm thanh khác với tốc độ tín hiệu điện tử.

- + Tốc độ lấy mẫu và cập nhật dữ liệu của các cảm biến là khác nhau.

Do vậy cần phải có khoảng thời gian và nhịp đồng hồ chung để đồng bộ hoá dữ liệu. Phải xây dựng các kỹ thuật để giải quyết bài toán độ trễ theo thời gian do quá trình truyền tín hiệu và phát hiện của cảm biến. Sự sắp xếp đảm bảo một khung thời gian chung.

- Các nền hoạt động của bộ cảm biến.

- + Đơn hoặc đa nền. Các cảm biến đơn hoặc đa nền cần phải đồng bộ hoá thời gian riêng của chúng.

- + Nền động hoặc tĩnh. Các cảm biến có thể hoạt động trên nền động hoặc tĩnh, bài toán sẽ trở nên phức tạp nếu các cảm biến hoạt động trên các nền động khác nhau. Ví dụ trên thực tế ta có thể sẽ không thu được kết quả có ý nghĩa khi có các cảm biến âm thanh trên nền chuyển động quá nhanh.

- Số lượng các cảm biến được sử dụng. Tăng số lượng các cảm biến dẫn đến hiện tượng tăng theo hàm số mũ của các yếu tố sau [65]:

- + Độ phức tạp khi thiết kế tổng hợp dữ liệu với thời gian thực.

- + Độ phức tạp của cấu trúc lưu trữ dữ liệu.

- + Chi phí truyền thông.

Nói chung, nếu có nhiều nguồn dữ liệu cảm biến thì sẽ có nhiều dữ liệu và sẽ không tòi về mặt lý thuyết để đưa ra quyết định. Tuy nhiên nhiều dữ liệu hoặc nhiều thông tin thì có thể dẫn đến tình huống khó đưa ra quyết định cuối cùng nếu các dữ liệu và thông tin đó không được tổ chức chặt chẽ và logic.

- Các chiến lược lựa chọn cảm biến. Việc lựa chọn cảm biến là nhằm mục đích lựa chọn được một cấu hình cảm biến thích hợp nhất từ các cảm biến có sẵn. Các yếu tố ảnh hưởng đến việc lựa chọn cảm biến gồm [79]:

- + Động học của đối tượng.
- + Các nguồn gây nhiễu.
- + Khả năng thực hiện của bộ cảm biến.

- Các lĩnh vực hoạt động. Quá trình tổng hợp dữ liệu mức cao cần phải xem xét đến lĩnh vực hoạt động của hệ thống là phục vụ dân sự hay quân sự.

- Phối hợp làm việc theo nhóm hoặc đơn lẻ. Để tạo ra được kết quả tổng hợp có ý nghĩa thì cần phải tìm hiểu chu trình luồng xử lý dữ liệu. Các nhóm các biến khác nhau sẽ tìm kiếm các thông tin khác nhau, sau đó hệ thống tổng hợp sẽ tổng hợp các tập dữ liệu và thông tin có ích từ các thông tin tìm kiếm được.

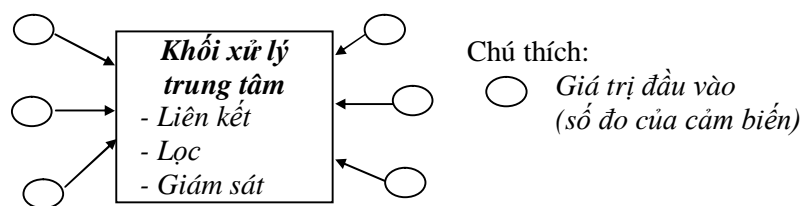
- Các hạn chế về mặt vật lý và môi trường hoạt động của cảm biến.

1.1.5. Kiến trúc hệ thống tổng hợp dữ liệu.

Ta có thể phân loại các hệ thống tổng hợp dữ liệu và thông tin thành 03 kiểu kiến trúc tổng hợp chính như sau [61]:

- Kiến trúc tổng hợp trung tâm.
- Kiến trúc tổng hợp cục bộ.
- Kiến trúc tổng hợp phân tán.

1.1.5.1. Kiến trúc tổng hợp trung tâm.



Hình 1.6 Kiến trúc trung tâm với một trung tâm xử lý.

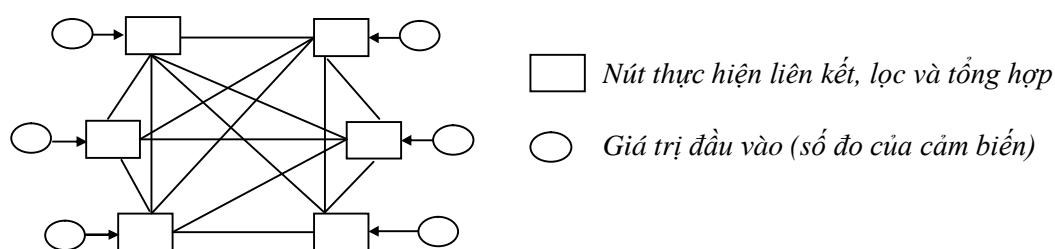
Trong kiến trúc tổng hợp trung tâm, mỗi một thao tác tổng hợp sẽ được thực hiện tại bộ xử lý trung tâm. Trung tâm sẽ nhận tất cả các thông tin từ nhiều nguồn cảm biến khác nhau (hình 1.6) và đưa ra các quyết định.

1.1.5.2. Kiến trúc tổng hợp phân tán.

Một kiến trúc tổng hợp phân tán sẽ bao gồm các nút mạng có những phương tiện xử lý riêng. Kiến trúc tổng hợp phân tán có 3 đặc điểm sau [40]:

- Không có trung tâm xử lý.
- Không có phương tiện giao tiếp chung.
- Các nút không cấu trúc toàn cục về liên kết mạng.

Kiến trúc tổng hợp phân tán có thể phân loại tiếp thành kiến trúc tổng hợp phân tán có liên kết hoàn chỉnh và kiến trúc tổng hợp phân tán có liên kết không hoàn chỉnh. Hình 1.7 dưới đây chỉ ra một kiến trúc tổng hợp phân tán có liên kết hoàn chỉnh.



Hình 1.7 Kiến trúc tổng hợp phân tán.

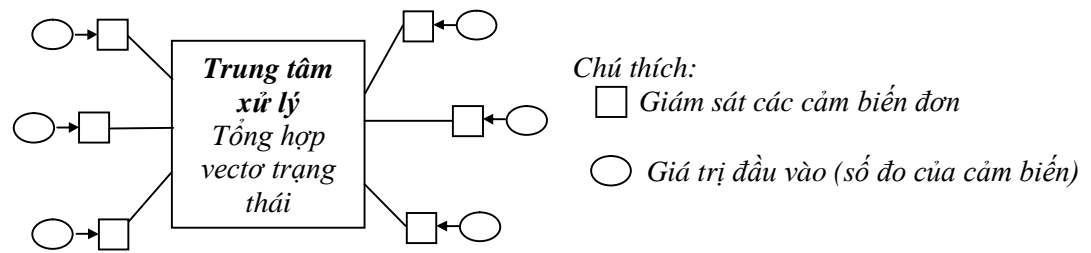
1.1.5.3. Kiến trúc tổng hợp cục bộ.

Kiến trúc tổng hợp cục bộ được chỉ ra trên hình 1.8. Số đo của mỗi một cảm biến sẽ được xử lý độc lập trước khi gửi tới trung tâm xử lý cho quá trình tổng hợp về sau với những nguồn đầu vào khác.

Kiến trúc tổng hợp cục bộ có thể xem như là một trường hợp mở rộng của kiến trúc tập trung.

Ngoài ra, trong một số trường hợp người ta cũng sử dụng các kiến trúc hỗn hợp kết hợp các kiến trúc tập trung, phân tán và cục bộ.

Để lựa chọn kiến trúc mạng phù hợp ta phải cân nhắc đến các yêu cầu của bài toán, cơ sở dữ liệu sẵn có, khả năng đáp ứng của công nghệ... Nói chung, kiến trúc tập trung được cài đặt và sử dụng nhiều hơn kiến trúc cục bộ. Phần lớn các hệ thống tổng hợp dữ liệu hiện nay theo cách này hay cách khác đều sử dụng kiến trúc tập trung. Kiến trúc cục bộ rất khó cài đặt do các hạn chế truyền thông và tính toán, chi phí và các vấn đề không gian thời gian.



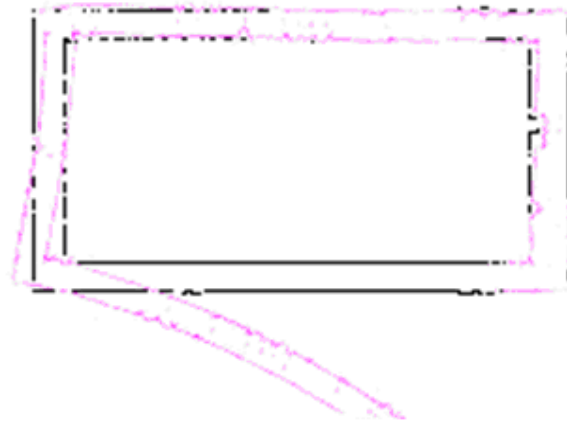
Hình 1.8 Kiến trúc tổng hợp cục bộ.

Phần lớn các nền tính toán độc lập như của robot di động, xe tự hành, ô tô, máy bay... đều hoạt động theo kiến trúc tập trung. Ở đây, nền tính toán độc lập là hệ thống chỉ có một bộ xử lý trung tâm. Việc chuyển các nền tính toán độc lập thành kiến trúc tổng hợp cục bộ phụ thuộc rất nhiều vào việc các ràng buộc được giải quyết như thế nào.

1.2. Các phương pháp tổng hợp cảm biến cho robot di động.

1.2.1. Định vị và lập bản đồ.

Để trả lời hai câu hỏi: robot đang ở đâu ? và sẽ đi đến đâu? là các phép định vị và lập bản đồ đường đi của robot. Cả hai nhiệm vụ thường liên quan chặt chẽ, bởi vì một robot không thể tạo ra một bản đồ chính xác nếu nó không biết đang ở đâu. Xét một ví dụ bản đồ được lập với Robot Nomad 200 như hình 1.9 [88]. Hành lang làm một đường vòng quanh hoàn chỉnh xung quanh trung tâm của tòa nhà. Đường nét đậm là đường thực của hành lang, và đường nét nhạt cho thấy hành lang được nhận biết bởi robot di động. Các cảm biến của robot di động cập nhật từng phần của hành lang vào bản đồ trong khi di chuyển. Trong trường hợp này, nó sử dụng bộ lập mã trục quay để xác định vị trí đã di chuyển. Và như đã biết, sai số tích lũy sẽ càng lớn nếu robot chạy trên một quãng đường càng dài (nguyên nhân là do kết cấu cơ khí không cân xứng giữa 2 bánh xe, ma sát trục quay, ma sát giữa từng bánh xe với mặt sàn, các thông số không thật hoàn toàn giống nhau của mỗi động cơ điện và hộp giảm tốc. v.v.), như biểu diễn trên hình 1.9.



Hình 1.9 Sự trôi phép định vị được phát sinh bởi robot Nomad 200. [88]

Phát triển một mô hình cho phép để ước tính sai số cho phép đo này thường rất khó khăn. Một số công nghệ mới, đặc biệt là GPS, INS có thể bù trừ các sai lệch này. Tuy nhiên GPS chỉ hoạt động đáng tin cậy ngoài trời với sai số thông thường từ vài đến chục mét. Các tín hiệu thường là không thể hoạt động tốt ở trong nhà, trong đường hầm, hoặc ở các thành phố với các tòa nhà lớn (đôi khi được gọi là hẻm núi đô thị).

Các nhà nghiên cứu đã cố gắng để giải quyết vấn đề định vị theo một số cách khác nhau. Phương pháp tiếp cận đầu tiên chỉ đơn giản là bỏ qua các sai số định vị. Trong khi phương pháp này có ưu điểm là đơn giản, nó loại bỏ việc sử dụng các phương pháp lập kế hoạch dẫn đường toàn cục. Ở đây giống như mô hình phản ứng [88], robot sẽ tìm đường đi về đích, gặp vật cản thì tránh và luôn luôn hướng về đích, như các thuật toán Bug; phương pháp trường thế (PFM); phương pháp tọa độ cực (VFH)[31];... Tuy nhiên những phương pháp này chỉ áp dụng cho dẫn đường ở bản đồ cục bộ, ít vật cản, và đôi khi robot không thể tìm về được đích và việc cho robot biết chính xác đích cũng là một bài toán khó. Một cách tiếp cận khác là phương pháp sử dụng dữ liệu bản đồ. Có 2 cấu trúc dữ liệu bản đồ: biểu diễn theo hình học và theo hình trạng. Loại *biểu diễn hình học* dựa trên hình dạng tuyệt đối của các đối tượng. Nó có thể là các bản đồ lưới chiếm giữ, bản đồ đường hoặc bản đồ các đa giác...[88,111]. Ngược lại, phương pháp *biểu diễn hình trạng* dựa trên các mối quan hệ hình học giữa các đặc trưng (đặc điểm) quan sát được chứ không phải là vị trí tuyệt đối của chúng đối với một hệ tọa độ bất kỳ. Kết quả là bản

đồ có dạng là một đồ thị, tại đó các nút biểu diễn cho các đặc tính được quan sát và các cạnh biểu diễn cho các mối quan hệ giữa các đặc điểm. Không giống như bản đồ hình học, bản đồ hình trạng có thể được xây dựng và duy trì mà không có bất kỳ dự đoán cho vị trí của robot. Điều này có nghĩa rằng các sai số trong biểu diễn này sẽ độc lập với bất kỳ sai số trong dự toán cho các vị trí robot Taylor [118]. Tuy nhiên, cách định vị này phụ thuộc nhiều vào một số điểm như các cửa ra vào, các vật mốc (landmark),... Đó là các vật tự nhiên hay nhân tạo ở các tọa độ biết trước và có những điểm đặc biệt để robot có thể nhận ra. Các vật mốc thụ động như các cửa ra vào trong nhà cần được chiếu sáng mới nhìn thấy được, và khả năng nhầm lẫn cửa này với một cửa khác (ví dụ, nó sẽ hiểu chỗ giao nhau với một hành lang như một cánh cửa...). Một vấn đề quan trọng nhất trong phương pháp định vị này là khả năng ước tính vị trí ban đầu và hướng rất kém (ví dụ, nếu robot được đặt trong một tòa nhà văn phòng nói với nó phải đối mặt với hướng Bắc và khi phải đối mặt với hướng Nam, nó sẽ phải hiệu chỉnh sau khi nó bất ngờ gặp phải một hoặc nhiều công) [88]. Vì vậy, hướng tiếp cận này rất ít được sử dụng. Hướng tiếp cận khác, rất phổ biến trong thực tế là các phương pháp sử dụng vài dạng định vị và lập bản đồ liên tục.

Về cơ bản robot di chuyển một khoảng cách ngắn và làm khớp những gì nó nhìn thấy với những gì nó đã xây dựng trong bản đồ của mình. Bản đồ làm khớp được thực hiện phức tạp hơn bởi độ bất định trong lưới chiếm giữ của chính nó. Những gì robot nghĩ rằng nó đã được nhìn thấy tại thời điểm t_{n-1} có thể đã sai và các quan sát tại thời điểm t_n là tốt hơn. Cách tiếp cận này rất chính xác, cho phép tích hợp một khu vực rộng lớn bản đồ mà không bị sai số tích lũy *dead-reckoning* vì tất cả các kết nối là tương đối giữa các nút thay vì tuyệt đối. Sau khi bản đồ đã được thiết lập, quá trình định vị cơ bản là quá trình làm khớp một bản đồ cục bộ đến vị trí thích hợp trên bản đồ được lưu trữ. Tuy nhiên, vấn đề quan trọng và đầy thách thức cho kỹ thuật định vị bản đồ này là việc phát triển những giải thuật tìm kiếm sự làm khớp giữa hai bản đồ (bản đồ cục bộ quanh vị trí hiện tại của robot và bản đồ toàn cục mà nó đã lưu trong bộ nhớ). Những giải thuật này thường được chia thành

2 loại: loại làm khớp *dựa trên hình tượng* (icon-based map matching) và loại làm khớp *dựa trên đặc điểm* (feature-based map matching) [88,111] và Talluri & Aggarwal [119].

Sự định vị có thể sử dụng trực tiếp dữ liệu cảm biến thô (*dựa trên hình tượng*) hoặc sử dụng trích chọn nét đặc trưng từ dữ liệu cảm biến (*dựa trên đặc điểm*).

Các thuật toán làm khớp *dựa trên hình tượng* thường xuất hiện phổ biến hơn trong thực tế, một phần bởi vì chúng thường sử dụng lưới chiếm giữ. Lưới chiếm giữ là một cơ chế cho đầu vào tổng hợp dữ liệu cảm biến vào một mô hình hoặc bản đồ toàn cục; hay là sẽ phù hợp với chỉ số đọc cảm biến hiện tại để tổng hợp dữ liệu với các phép đo cảm biến trước đó trong lưới chiếm giữ.

Sự định vị *dựa trên đặc điểm* đầu tiên có thể trích xuất một đường viền từ dữ liệu siêu âm hoặc lưới chiếm giữ, sau đó trên cơ sở thu thập dữ liệu tiếp theo, robot sẽ trích xuất đường viền và tính toán vị trí thay đổi thực sự. Nhìn chung, làm khớp được thực hiện bởi các trích chọn nét đặc trưng đầu tiên, tiếp theo bằng cách xác định sự tương ứng chính xác giữa các nét đặc trưng hình ảnh và mô hình, thường là do một số hình thức tìm kiếm ràng buộc Cox [33]. Sự định vị này có khái niệm tương tự như ý tưởng các *địa điểm đặc biệt* trong dẫn đường hình trạng, theo nghĩa là có các nét đặc trưng trong môi trường có thể được nhìn thấy từ một số quan điểm Kak [72].

Shaffer et al. [112], so sánh các phương pháp *dựa trên hình tượng* và *dựa trên đặc điểm*. Tác giả đã kết luận rằng, các phương pháp *dựa trên hình tượng* ít chịu hạn chế hơn đối với môi trường và có độ chính xác hơn so với các phương pháp *dựa trên đặc điểm* (bởi vì, các phương pháp *dựa trên đặc điểm* phải chọn môi trường phù hợp; biết các loại nét đặc trưng để mà trích chọn, các điểm dữ liệu lấy mẫu ít hơn nên độ chính xác ít hơn). Tuy nhiên, các thuật toán *dựa trên đặc điểm* thường chạy nhanh hơn vì có ít dữ liệu hơn để làm khớp bản đồ trong quá trình định vị.

Nhiều nhà nghiên cứu tin rằng chưa có một thuật toán chung và không có phương thức cảm biến duy nhất có thể nắm bắt tất cả tính năng có liên quan của một môi trường thực tế cho bài toán định vị. Do đó cần nhiều phương pháp hay thuật toán tổng hợp cảm biến để ước tính chính xác vị trí robot hơn. Dưới đây là một vài ví dụ:

- Buchenberger M., et al. [27] đã phát triển một cơ chế sử dụng thông tin không đồng nhất thu được từ một laser-radar và hệ thống sonar để xây dựng một mô hình toàn cục đáng tin cậy và đầy đủ.

- Courtney và Jain [34] tích hợp ba nguồn cảm biến thông thường (sonar, thị giác, và hồng ngoại) cho một sự biểu diễn không gian dựa trên cảm biến. Các tác giả thực hiện một cách tiếp cận mức - đặc trưng để tổng hợp cảm biến từ các bản đồ lưới đa cảm biến nhận được sử dụng một phương pháp toán học dựa trên những khoảng khắc không gian và thời điểm bất biến.

- Moravec Hans P. [82], [99] và Yenilmez Levent [131] sử dụng kỹ thuật *ô chia lưới (certainty grid)* với xác suất *ô lưới chiếm chỗ* để xây dựng được bản đồ dẫn đường cho robot di động việc tính toán các giá trị ô lưới chiếm chỗ thường thực hiện bằng suy luận Bayesian, tuy nhiên thông tin cảm biến thường là không hoàn hảo, hạn chế tầm phủ với thời gian và không gian, độ chính xác bị giới hạn, sự cố và hoặc các phép đo mơ hồ của cảm biến. Để tăng khả năng và hiệu suất thường phải sử dụng thêm các loại thiết bị cảm biến khác nhau để bổ sung cho nhau, như vậy việc lập bản đồ với các phép đo cảm biến khác nhau để tính toán đưa vào ô lưới là một vấn đề cần giải quyết [80].

- Matthies và Elfes [89] đề xuất một phương pháp thống nhất cho việc tổng hợp các loại cảm biến khác nhau. Mỗi bộ cảm biến được gán vào một mô hình thể hiện không gian, được phát triển cho từng loại cảm biến, mà các bản đồ phép đo cảm biến đưa vào tương ứng các ô. Như vậy rất khó, bởi vì thời gian xử lý của mỗi phép đo cảm biến khác nhau, độ bất định mỗi cảm biến khác nhau khó có thể đồng bộ để đưa vào tính toán.

- Theo Chenavier & Crowley [35] để làm khớp chắc chắn các dữ liệu nhận được hiện tại với mô hình môi trường được lưu trữ, một số đặc trưng phải được sử dụng cùng một lúc. Điều này đặc biệt đúng đối với một vùng hệ thống dựa trên hình ảnh từ đó các loại điểm đặc trưng được giới hạn với một vùng bản đồ hình ảnh. Nhìn chung, càng nhiều điểm đặc trưng được sử dụng trong khi làm khớp thì ít có khả năng xảy ra lỗi làm khớp, nhưng nó phải mất thêm thời gian để xử lý.

1.2.2. Tổng hợp cảm biến với phương pháp suy luận xác suất để nâng cao độ tin cậy cho bài toán định vị robot.

Phương pháp suy luận xác suất Bayesian [49][80][82][90] là một thuật toán tổng hợp dữ liệu thống kê dựa trên định lý Bayes [120] với xác suất có điều kiện hay xác suất hậu nghiệm để ước tính vector trạng thái n-chiều 'X', sau khi đã được quan sát hoặc đo được hàm 'Z'. Thông tin ngẫu nhiên chứa trong Z và X được mô tả bởi một hàm mật độ xác suất (p.d.f) $p(Z/X)$, được gọi là *hàm khả năng*, hoặc mô hình cảm biến, đó là hàm mục tiêu phụ thuộc vào quan sát. Hàm khả năng liên quan đến mức độ mà xác suất hậu nghiệm bị thay đổi và được đánh giá thông qua các thí nghiệm độc lập hoặc bằng cách sử dụng các thông tin sẵn có về vấn đề này. Nếu thông tin trạng thái X được thực hiện độc lập trước khi quan sát, sau đó hàm khả năng có thể được cải thiện để đưa ra kết quả chính xác hơn. Như một thông tin tiên nghiệm về X có thể được đóng gói như xác suất tiên nghiệm $P(X = x)$ và được coi như là chủ quan bởi vì nó không dựa trên dữ liệu quan sát được. Định lý Bayes đưa ra một phân phối có điều kiện hậu nghiệm là $X = x$, với giá trị đo được $Z = z$.

$$p(X = x | Z = z) = \frac{p(Z = z | X = x)P(X = x)}{\int p(Z = z | X = x)P(X = x)dx} = \frac{p(Z = z | X = x)P(X = x)}{p(Z = z)} \quad (1.1)$$

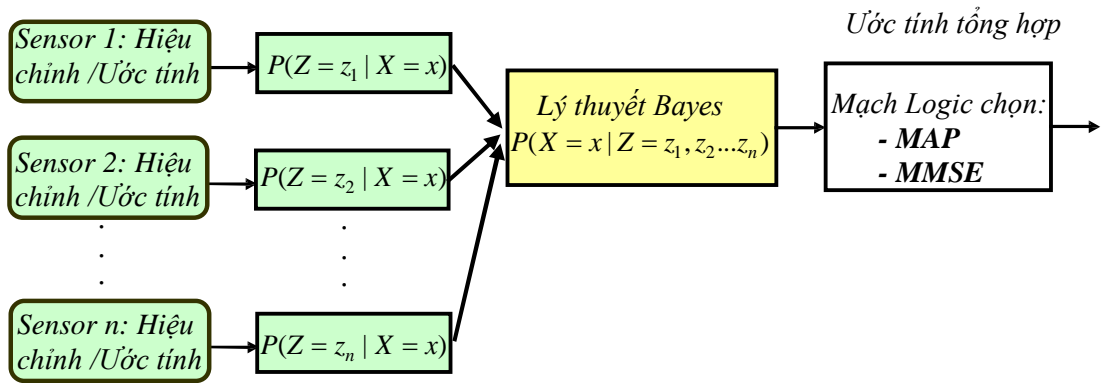
Do mẫu số chỉ phụ thuộc vào giá trị đo (tổng được thực hiện trên tất cả các giá trị có thể của trạng thái), một cách tiếp cận trực quan để ước tính có thể được thực hiện bằng cách lấy cực đại phân bố hậu nghiệm, ví dụ, bằng cách lấy cực đại số phân tử của (1.1). Điều này được gọi là *ước tính tối đa một hậu nghiệm - MAP (Maximum a posteriori)*, và được đưa ra bởi:

$$\hat{x}_{MAP} = \arg \max p(X = x | Z = z) \propto p(Z = z | X = x)P(X = x) \quad (1.2)$$

Một kiểu ước lượng thông thường khác được gọi là hàm ước lượng *sai số bình phương trung bình tối thiểu* – *MMSE (Minimum Mean Square Error)* tối thiểu tổng bình phương của các sai số, nghĩa là, giảm thiểu khoảng cách Euclide giữa trạng thái thật và ước tính sau khi quan sát đã được thực hiện.

Thí dụ, để hợp nhất các giá trị đo từ hai cảm biến, (1.1) có thể viết lại như sau:

$$p(X = x | Z = z_1, z_2) = \frac{p(Z = z_1 | X = x)p(Z = z_2 | X = x)P(X = x)}{p(Z = z_1, z_2)} \quad (1.3)$$



Hình 1.10 Tổng hợp đa cảm biến sử dụng kỹ thuật Bayesian. [53]

1.2.3. Tổng hợp cảm biến bằng bộ lọc Kalman để nâng cao độ tin cậy cho bài toán định vị robot.

Bộ lọc Kalman rời rạc sử dụng mô hình toán học cho lọc các tín hiệu bằng cách sử dụng các phép đo với một số lượng đáng kể các sai số thống kê và có hệ thống. Phương pháp được phát triển bởi Kalman và Bucy vào năm 1960 [73,129].

Thông thường, một bộ lọc Kalman tổng hợp dữ liệu đo được trong một khoảng thời gian liên tiếp, cung cấp một ước tính khả năng xảy ra lớn nhất của một tham số. Nó cũng có thể liên kết với các đầu vào của đa cảm biến thành một vec-tơ của các trạng thái bên trong có chứa các tham số quan tâm, miễn là chúng chỉ phụ thuộc tuyến tính giữa các đầu vào và các trạng thái của hệ thống [121]. Bộ lọc sử dụng thuật toán thời gian rời rạc để loại bỏ nhiễu từ các tín hiệu của cảm biến để tạo

ra được dữ liệu tổng hợp, ví dụ ước tính các giá trị làm tròn của vị trí, vận tốc, và gia tốc tại một dãy các điểm trong quỹ đạo. Mô hình bộ lọc Kalman chuẩn được suy ra từ 2 phương trình tuyến tính.

Phương trình thứ nhất mô tả động học của hệ thống:

$$\mathbf{x}_k = A.\mathbf{x}_{k-1} + B.\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \quad (1.4)$$

Trong đó \mathbf{x}_k là véc tơ trạng thái của hệ thống tại thời điểm k . A là ma trận trạng thái không suy biến. Véc-tơ \mathbf{u}_{k-1} miêu tả đầu vào hệ thống tại thời điểm $k-1$. Mỗi quan hệ véc-tơ đầu vào \mathbf{u}_{k-1} và véc-tơ trạng thái \mathbf{x}_k được xác định bằng ma trận B , \mathbf{w}_{k-1} là một biến ngẫu nhiên biểu diễn cho nhiễu hệ thống.

Phương trình tuyến tính thứ 2 mô tả quan sát nhiễu của hệ thống, với phép đo $\mathbf{z} \in \mathfrak{R}^m$:

$$\mathbf{z}_k = H.\mathbf{x}_k + \mathbf{v}_k \quad (1.5)$$

Trong mỗi phần tử của véc-tơ \mathbf{z}_k chứa một quan sát cảm biến tại thời điểm k . Ma trận H liên quan tới các phép đo trạng thái bên trong, \mathbf{v}_k biểu diễn nhiễu đo.

Các biến ngẫu nhiên \mathbf{w}_k và \mathbf{v}_k được giả định là nhiễu quá trình và nhiễu đo; độc lập với nhau, ồn trắng và có phân bố xác suất chuẩn:

$$\mathbf{w}_k \sim \mathbf{N}(0, Q_k) \quad \mathbf{v}_k \sim \mathbf{N}(0, R_k) \quad E(\mathbf{w}_i \mathbf{v}_j^T) = 0$$

Q_k là các ma trận hiệp phương sai nhiễu đầu vào; R_k là ma trận hiệp phương sai của nhiễu đo.

Bài toán bộ lọc Kalman chính là tìm giá trị ước lượng của trạng thái \mathbf{x} khi biết được sự biến thiên của nó mà ta đo được một đại lượng \mathbf{z} phụ thuộc tuyến tính vào \mathbf{x} . Hay nói cách khác, dựa trên các số liệu đo \mathbf{z} , có thể tìm được một hệ số Kalman K trong mỗi chu kỳ lấy mẫu tín hiệu đo thuộc một vòng lặp đệ quy gọi là bộ lọc Kalman, sao cho giá trị ước lượng trạng thái của hệ gần với giá trị thực nhất.

Như vậy, quá trình thực hiện của bộ lọc Kalman là một vòng lặp đệ quy gồm 2 giai đoạn là pha dự đoán (với các phương trình cập nhật thời gian) và pha hiệu chỉnh (với các phương trình cập nhật số liệu). Hệ số khuếch đại Kalman K_k sẽ được

tính toán, sau đó véc-tơ trạng thái và ma trận lỗi tương quan P_k sẽ được cập nhật từ các đánh giá bước trước $\hat{\mathbf{x}}_k^-$ và P_k^- .

Để khởi động bộ lọc Kalman, người ta phải cung cấp một ước tính tiên nghiệm $\hat{\mathbf{x}}_0^-$ và một ước tính hiệp phương sai lỗi của nó P_0^- . P_0^- có thể được khởi tạo với một ước tính giá trị ban đầu không chính xác, khi đó ứng dụng tiếp theo của bộ lọc Kalman sẽ cho P_k^- tiếp cận gần với giá trị chính xác của nó.

Hình 1.11 trình bày sơ đồ thuật toán lọc Kalman. Trong pha dự đoán, các cập nhật thời gian cho ước lượng trạng thái tiên nghiệm và hiệp phương sai sai số tiên nghiệm được tiến hành:

$$\hat{\mathbf{x}}_k^- = A \hat{\mathbf{x}}_{k-1} + B \mathbf{u}_{k-1} \quad (1.6)$$

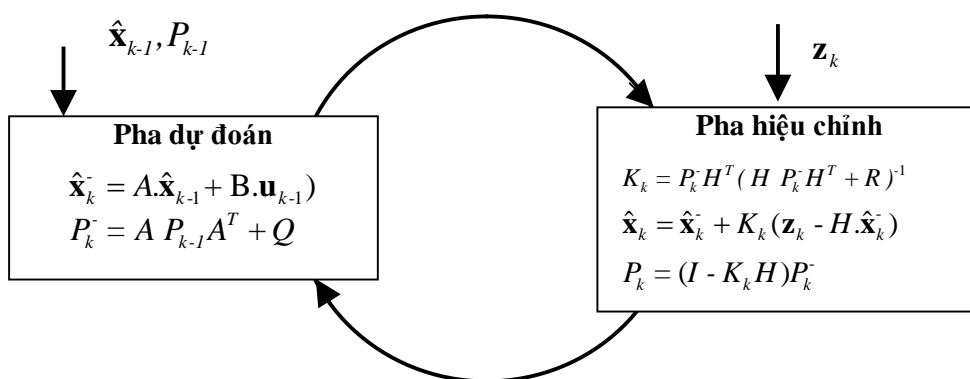
$$P_k^- = A P_{k-1} A^T + Q \quad (1.7)$$

Trong pha hiệu chỉnh, hệ số Kalman sẽ được tính và ước lượng trạng thái hậu nghiệm được cập nhật trên cơ sở phép đo \mathbf{z} :

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (1.8)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - H \hat{\mathbf{x}}_k^-) \quad (1.9)$$

$$P_k = (I - K_k H) P_k^- \quad (1.10)$$



Hình 1.11 Sơ đồ thuật toán bộ lọc Kalman chuẩn rời rạc. [129]

Nếu giá trị của Q được chọn là lớn hơn chính giá trị thật của nó, bộ lọc Kalman sẽ đặt độ tin cậy vào các giá trị đo lường hơn là hệ thống. Khi đó, các kết quả đánh giá sẽ phụ thuộc sai số của phép đo. Nếu giá trị Q chọn nhỏ hơn chính nó,

độ trễ thời gian sẽ được tăng lên. Khi Q nhỏ hơn rất nhiều so với giá trị thật, bộ lọc sẽ phân kỳ dẫn đến kết quả là không ổn định về mặt tính toán, như vậy việc chọn giá trị Q thông thường dựa trên thực nghiệm.

Trong robot di động, bộ lọc Kalman được sử dụng để hiệu chỉnh định vị dựa trên hai hoặc nhiều dữ liệu đầu vào khác nhau [36,51,121,134]. Tuy nhiên như đã mô tả phần trên, bộ lọc Kalman giải quyết các vấn đề phổ biến của ước tính trạng thái $\mathbf{x} \in \mathfrak{R}^m$ mà quá trình điều khiển thời gian rời rạc phụ thuộc vào phương trình sai phân ngẫu nhiên tuyến tính. Nhưng điều gì sẽ xảy ra, nếu quá trình ước tính hoặc mối quan hệ giữa các phép đo với quá trình điều khiển là phi tuyến? Giải quyết bài toán này là một bộ lọc Kalman được tuyến tính hóa hoặc bộ lọc Kalman mở rộng viết tắt là EKF (*extended Kalman filter*). Bằng việc khai triển chuỗi Taylor các hàm hệ thống và hàm đo phi tuyến, ta có thể tuyến tính hóa sự dự đoán xung quanh dự đoán hiện tại sử dụng một phần dẫn xuất của quá trình và chức năng đo lường để đưa vào tính toán dự đoán. Để thực hiện, ta bắt đầu hiệu chỉnh từ bộ lọc Kalman rời rạc. Hãy giả định rằng phương trình trạng thái của hệ thống (1.4) của ước tính trạng thái $\mathbf{x} \in \mathfrak{R}^m$, nhưng quá trình điều khiển thời gian rời rạc phụ thuộc vào phương trình sai phân ngẫu nhiên phi tuyến tính:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \quad (1.11)$$

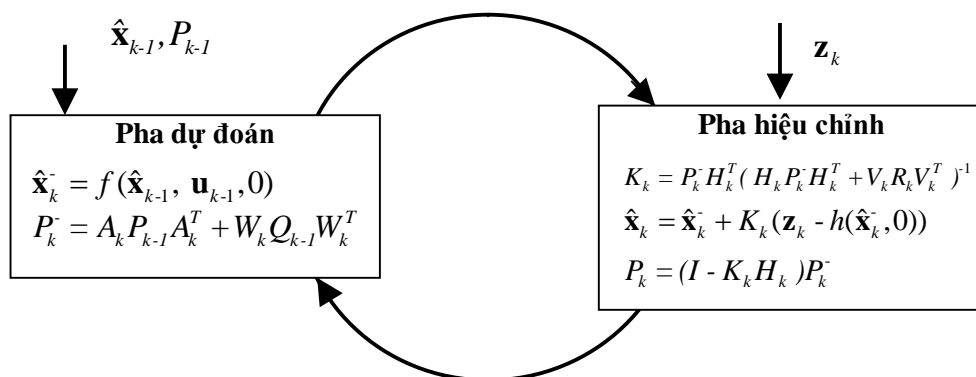
Và phép đo $\mathbf{z} \in \mathfrak{R}^m$:

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (1.12)$$

Các biến ngẫu nhiên \mathbf{w}_k và \mathbf{v}_k cũng được giả định là nhiễu quá trình và nhiễu đo như (1.4) và (1.5), trong trường hợp này hàm phi tuyến f ở (1.11) liên quan đến trạng thái ở bước thời gian hiện tại k đến trạng thái ở bước thời gian trước đó $k-1$. Nó còn bao gồm các tham số đầu vào \mathbf{u}_k và nhiễu quá trình \mathbf{w}_k . Hàm phi tuyến h của phép đo trong phương trình (1.12) biểu diễn mối liên quan giữa trạng thái \mathbf{x}_k với phép đo \mathbf{z}_k .

Tương tự như bộ lọc Kalman chuẩn rời rạc, quá trình thực hiện của bộ lọc Kalman mở rộng cũng là một vòng lặp đệ quy với 2 giai đoạn là pha dự đoán (với

các phương trình cập nhật thời gian) và pha hiệu chỉnh (với các phương trình cập nhật số liệu) như trên hình 1.12.



Hình 1.12 Sơ đồ thuật toán bộ lọc Kalman mở rộng [129].

1. *Pha dự đoán* với các phương trình cập nhật:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0) \quad (1.17)$$

$$P_k^- = A_{k-1} P_{k-1} A_{k-1}^T + W_{k-1} Q_{k-1} W_{k-1}^T \quad (1.18)$$

2. *Pha hiệu chỉnh* với các phương trình cập nhật phép đo:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (1.19)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0)) \quad (1.20)$$

$$P_k = (I - K_k H_k) P_k^- \quad (1.21)$$

Trong đó:

$\hat{\mathbf{x}}_k^- \in \mathcal{R}^n$ là ước lượng trạng thái tiên nghiệm ở bước k nhận được từ tiên nghiệm quá trình ở bước k .

$\hat{\mathbf{x}}_k \in \mathcal{R}^n$ là ước lượng trạng thái hậu nghiệm ở bước k nhận được sau phép đo \mathbf{z}_k .

P_k^- là ma trận hiệp biến của sai số ước lượng trạng thái tiên nghiệm.

P_k là ma trận hiệp biến của sai số ước lượng trạng thái hậu nghiệm.

Q_{k-1} là ma trận hiệp phương sai nhiễu đầu vào.

R_k là ma trận hiệp phương sai của nhiễu đo.

K_k là hệ số lọc Kalman.

I là ma trận đơn vị.

Các biến ngẫu nhiên \mathbf{w}_k và \mathbf{v}_k được giả định là nhiễu quá trình và nhiễu đo như (1.13) và (1.14).

- A là ma trận Jacobian của các đạo hàm riêng của f theo \mathbf{x} :

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0).$$

- W là ma trận Jacobian của các đạo hàm riêng của f theo \mathbf{w} :

$$W_{ij} = \frac{\partial f_i}{\partial w_j}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0).$$

- H là ma trận Jacobian của các đạo hàm riêng của h theo \mathbf{x} :

$$H_{ij} = \frac{\partial h_i}{\partial x_j}(\hat{\mathbf{x}}_k, 0).$$

- V là ma trận Jacobian của các đạo hàm riêng của h theo \mathbf{v} :

$$V_{ij} = \frac{\partial h_i}{\partial v_j}(\hat{\mathbf{x}}_k, 0).$$

Qua những phân tích ở trên chúng ta thấy rằng khi sử dụng bộ lọc Kalman rời rạc cho phép đạt được nhiều kết quả mong muốn [119]. Tuy nhiên, mô hình tuyến tính của hệ thống không phải lúc nào cũng khả thi. Vì vậy EKF đã được dẫn xuất từ các phương trình ngẫu nhiên phi tuyến cho mô hình hệ thống [68][129].

Việc ứng dụng EKF cho robot di động rất đa dạng, ví dụ như định vị cho robot di động có các bánh xe được điều khiển vi sai [116], hay ứng dụng EKF cho tổng hợp đa cảm biến để nâng cao khả năng ước lượng trạng thái của robot di động [69,122]. Thêm một lợi thế trong một môi trường đa cảm biến với các ma trận đo lường giống hệt nhau, các phép đo có thể được xử lý một cách riêng biệt để có được độ phức tạp tính toán thấp hơn. Trong trường hợp các ma trận đo không giống nhau là thích hợp sử dụng hợp nhất véc-tơ đầu vào chứa các thông tin của tất cả các cảm biến. Thêm một lý do quan trọng khi tổng hợp đa cảm biến có các loại cảm biến khác nhau, do thời gian xử lý trên mỗi cảm biến khác nhau, thời gian cập nhật trên mỗi cảm biến khác nhau, nhưng vẫn có thể mô hình hóa được với các dạng thông tin của EKF, và do đó có thể tổng hợp được nhiều cảm biến khác nhau, tăng số chiều dữ liệu [54].

1.3. Kết luận.

Tổng hợp cảm biến đưa ra một giải pháp tuyệt vời để vượt qua các giới hạn vật lý của các hệ thống cảm biến. Nó cho phép giảm độ phức tạp của phần mềm quản lý, làm ẩn các thuộc tính của các cảm biến vật lý đằng sau lớp tổng hợp cảm biến.

Trong chương này, tác giả đã tổng quan các phương pháp tổng hợp cảm biến, cũng như áp dụng phương pháp tổng hợp cảm biến để định vị cho robot di động. Vấn đề định vị chính xác phụ thuộc vào các phép đo của cảm biến. Để nâng cao hiệu quả các phép đo của các cảm biến thì phương pháp tổng hợp cảm biến là phương pháp hiệu quả nhất được sử dụng trong robot hiện đại, từ đó một số phương pháp tổng hợp cảm biến đã được nêu ra. Tuy nhiên, hiện nay bộ lọc Kalman và suy luận Bayesian là những công cụ được sử dụng nhiều nhất trong tổng hợp các cảm biến.

CHƯƠNG 2

XÂY DỰNG MÔ HÌNH ROBOT DI ĐỘNG ĐA CẢM BIẾN

Để phục vụ việc nghiên cứu các phương pháp tổng hợp cảm biến cho dẫn đường robot di động ta cần có một mô hình robot di động đa cảm biến. Với mục đích này chúng tôi đã thiết kế chế tạo được một robot di động đa cảm biến có cấu trúc đặc điểm sau: ngoài *bộ điều khiển* và *bộ chấp hành*, *bộ cảm nhận* của robot sẽ được chọn lựa các cảm biến phù hợp để bù trừ ưu nhược điểm cho nhau; hệ thống truyền tin giữa các bộ phận phải có tốc độ đủ đáp ứng với yêu cầu thời gian thực. Từ xuất phát điểm như vậy các cảm biến đã được lựa chọn trang bị cho robot là: cảm biến lập mã trục quay (còn gọi là lập mã quang - *optical encoder*), cảm biến chỉ hướng từ - địa bàn (*compass sensor*), cảm biến đo xa siêu âm (*ultrasonic sensor*), cảm biến đo xa laser LRF (*laser range finder*) và cảm biến ảnh toàn phương (*omni-directional camera*). Mặc dù các cảm biến này đều là các thương phẩm, nhưng người sử dụng phải lập trình nhúng để có thể thu thập được số liệu. Ngoài ra để tăng tính nhận dạng của thiết bị laser LRF 2D, chúng tôi đã cải tiến thiết bị đo này thành một cảm biến đo xa 3D bởi một hệ thống cơ khí và phần mềm xử lý và điều khiển đặc biệt như sẽ được trình bày riêng ở phần sau.

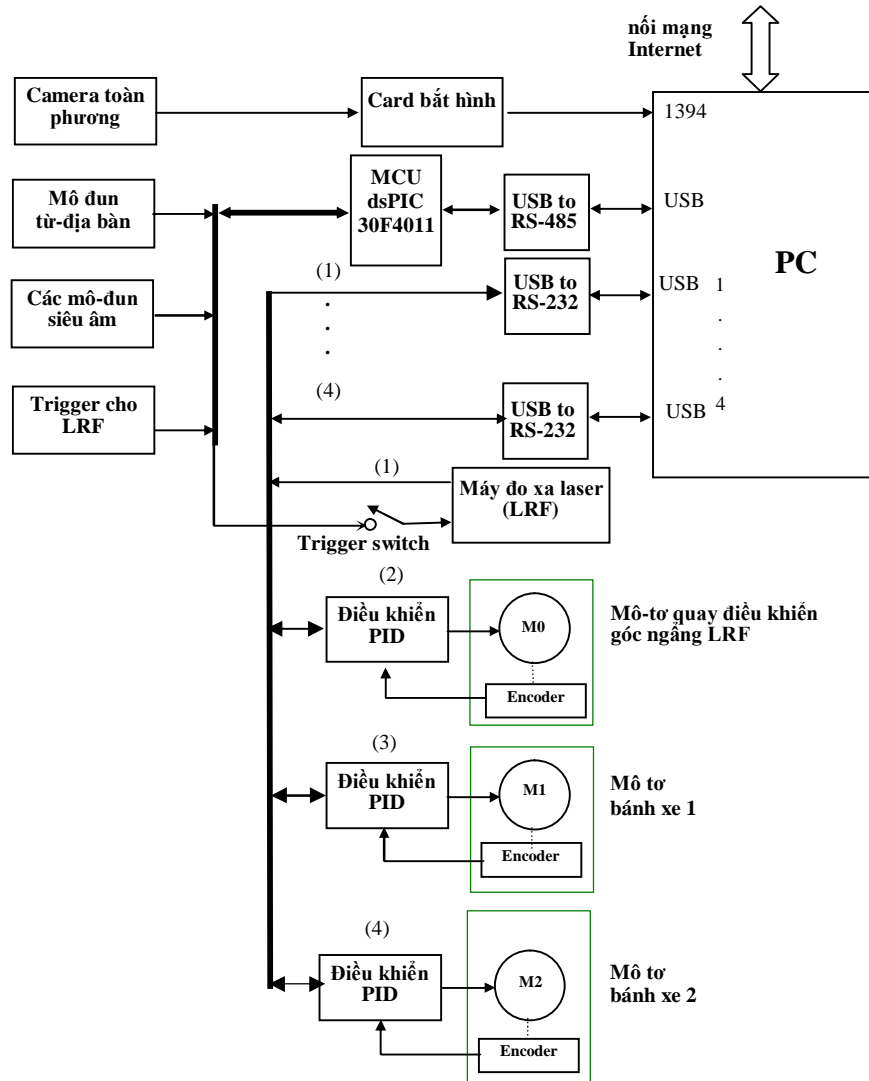
Nội dung chương này sẽ trình bày các bước thiết kế và chế tạo mô hình robot di động đa cảm biến tại phòng thí nghiệm.

2.1. Thiết kế chế tạo phần cứng mô hình robot di động đa cảm biến.

2.1.1. Các mô đun cảm nhận, khối điều khiển và cơ cấu chấp hành của robot.

Mạng thông tin cảm nhận của robot di động đa cảm biến được trình bày trên hình 2.1, trong đó máy vi tính là bộ phận điều khiển trung tâm. Do đặc điểm về tốc độ thông tin khác nhau giữa các cảm biến và thông tin ở đây là loại điểm- đa điểm, nên một thiết kế đặc biệt cho mạng truyền tin trong robot đã được thực hiện. Như thấy trên hình 2.1, cảm biến laser và các phần tử chấp hành của robot (các mô-tơ) được kết nối với *bộ phận điều khiển* của robot qua chuẩn ghép nối nối tiếp RS-232C; cảm biến camera toàn phương được kết nối với *bộ phận điều khiển* qua kênh

thông tin tốc độ cao IEEE-1394; các cảm biến còn lại (cảm biến từ địa bàn, các cảm biến siêu âm, công tắc hành trình) kết nối thông tin với bộ phận điều khiển qua chip vi điều khiển MCUdsPIC 30F4011. Việc kết nối giữa các cảm biến này với vi điều khiển được thực hiện qua chuẩn truyền thông công nghiệp với đường truyền điểm-đa điểm RS-485 theo giao thức hồi vòng Modbus/RTU.

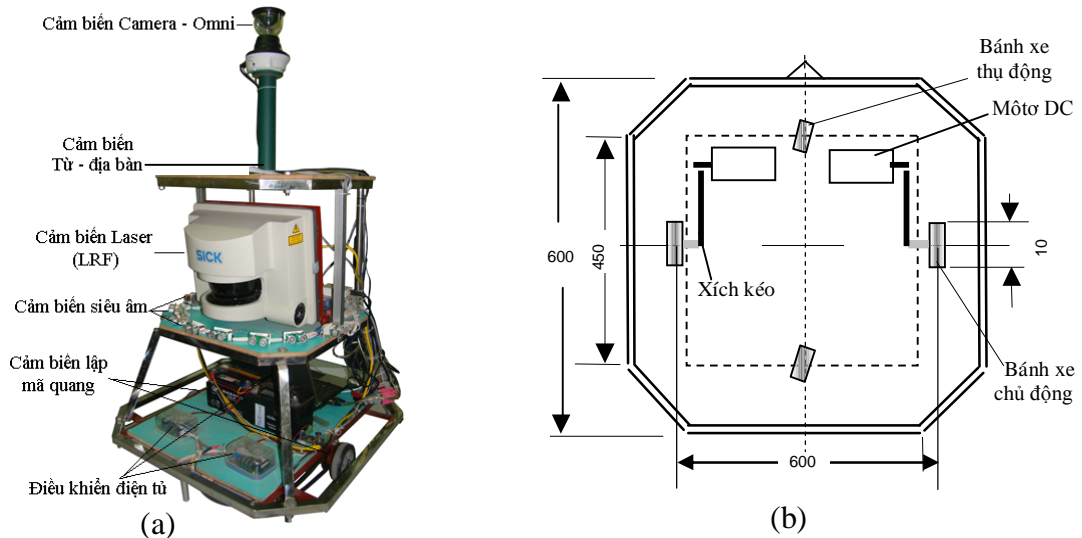


Hình 2.1 Mạng thông tin cảm nhận trong robot đa cảm biến.

Ngoài hệ thống các cảm biến nêu trên dùng cho hoạt động tự trị (autonomous), chúng tôi còn thiết kế hệ thống điều khiển robot từ xa qua mạng Internet 3G, cho phép người điều khiển có thể điều khiển robot bằng một cần điều khiển Joystick [100].

2.1.1.1. Cấu trúc cơ khí của robot.

Cấu trúc cơ khí và cơ cấu chấp hành của robot được thể hiện trên hình 2.2. Robot chuyển động được với 2 bánh xe vi sai, mỗi bánh được gắn với một mô-tơ điện điều khiển độc lập. Góc nghiêng của cảm biến LRF được điều khiển bằng một mô-tơ servo với cơ cấu truyền động quay cơ khí.



Hình 2.2 a) Hình ảnh của robot đa cảm biến được thiết kế chế tạo; b) chi tiết bản vẽ cơ khí để robot với 2 bánh xe chủ động cùng mô tơ.

- *Đặc điểm về cơ khí của robot:*

+ Kích thước đáy: 0,6 x 0,6m; Chiều cao: 1,2m.

+ Trọng lượng: 31kg.

+ Tốc độ tối đa: 0,450 m/s.

- *Đặc điểm về điện tử:*

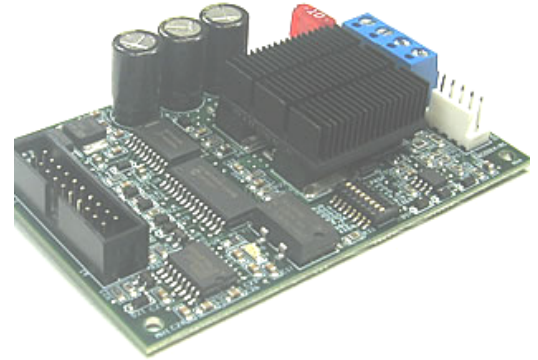
+ Hệ thống điều khiển mô tơ 2 bánh xe độc lập, và hệ thống điều khiển góc nghiêng cho cho LRF để tạo hình ảnh 3D.

+ 2 ắc qui 12V 9Ah cho thời lượng sử dụng trung bình 3 giờ.

+ 1 cảm biến LRF và 8 cảm biến siêu âm cùng nhiều cảm biến phục vụ cho các chức năng đặc biệt khác nhau như: cảm biến ảnh camera toàn phương, cảm biến từ - địa bàn, cảm biến lập mã quang, công tắc hành trình.

2.1.1.2. Các mô-đun điều khiển chuyển động.

Để điều khiển góc nghiêng của LRF và chuyển động của bánh xe robot chúng tôi sử dụng 3 mô-đun điều khiển mô-tơ “*Motion Mind*” thương phẩm. Đây là một mạch vi xử lý điều khiển độ rộng xung điện PWM (Pulsed Width Modulation) cấp cho mô-tơ theo luật PID (Proportional-Integral-Derivative Control). Phần mềm điều khiển được phát triển ở mức thấp (phần dẻo) nhưng



Hình 2.3 Mô-đun *Motion Mind* [83].

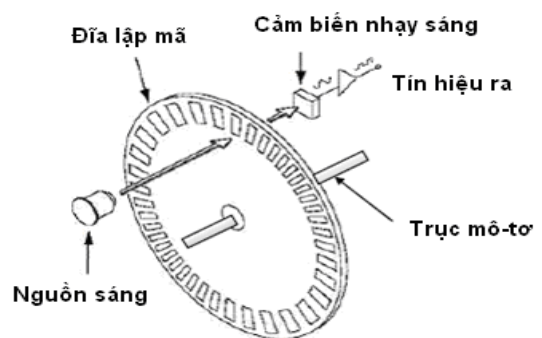
vào vi xử lý nhằm cho phép quá trình điều khiển sự ổn định tốc độ được diễn ra độc lập, không chiếm dụng thời gian của toàn chương trình điều khiển hệ thống robot.

Mô-đun *Motion Mind* có dòng điện cung cấp liên tục từ 9A÷25A điện áp làm việc 6V÷32V; được kết nối với máy tính qua chuẩn ghép nối tiếp RS-232C. Chương trình điều khiển được chúng tôi phát triển trên C++.

2.1.1.3. Các mô-đun cảm biến trên robot.

- *Cảm biến lập mã trục quay (encoder).*

Cảm biến được gắn trực tiếp với trục quay mô-tơ, dùng để đo vận tốc 2 bánh xe robot. Bộ lập mã trục quay như trình bày trên hình 2.4 về cơ bản là một đĩa quay làm đóng-ngắt chùm tia sáng đi qua các khe. Cùng bộ phận điện tử chuyển đổi quang-điện (optron), cơ cấu này phát ra số xung điện lồi ra tương ứng với một vòng quay của trục mô-tơ. Nếu đường kính của bánh xe và tỷ số truyền lực (tỷ số truyền bánh răng từ trục mô-tơ ra bánh xe robot) đã biết thì vị trí góc và tốc độ quay của



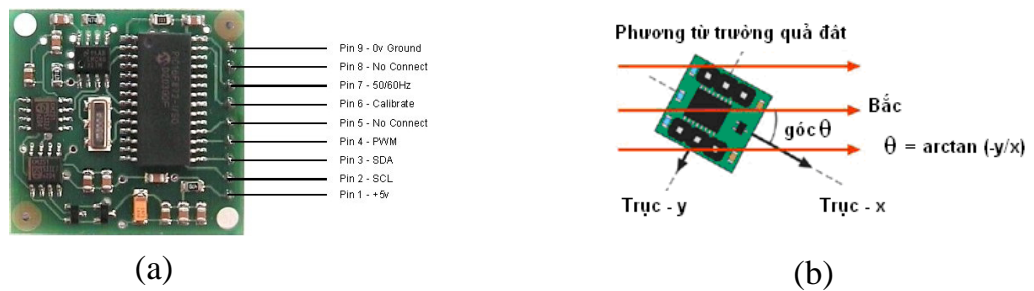
Hình 2.4 Cảm biến lập mã quang.

bánh xe có thể xác định được bằng việc đếm số xung ánh sáng chiếu qua các khe hẹp của đĩa lập mã. Việc xác định vị trí của robot bằng bộ lập mã này là phương

pháp phổ biến trên thế giới có tên gọi là phương pháp *Odometry* [22,50]. Phương pháp này có ưu điểm đơn giản dễ sử dụng để đo vận tốc của bánh xe. Tuy nhiên nó có nhược điểm là mắc phải sai số tích lũy, gây nên sự bất định của vị trí ước tính bởi hệ thống odometric tăng theo thời gian trong khi robot di chuyển. Thường phải triệt sai số tích lũy này một cách định kỳ bằng hỗ trợ bởi một cảm biến khác [50]. Trong hệ thống của chúng tôi đĩa lập mã quang có 500 khe sáng/1 đĩa.

- *Cảm biến chỉ hướng (heading sensor).*

Cảm biến cho phép đo góc hướng của robot so với phương từ trường của quả đất. Cảm biến được tích hợp trong một mô-đun cảm biến từ-địa bàn CMPS03 của hãng Philips (trên hình 2.5a) hoạt động dựa trên nguyên tắc hiệu ứng Hall với độ phân giải góc là $0,1^\circ$. Nguyên tắc hoạt động như biểu diễn trên hình 2.5b, mô-đun có 2 trục hướng *X* và *Y*.



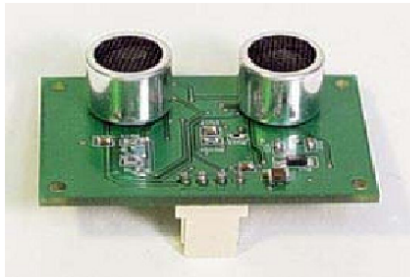
Hình 2.5 a) Mô-đun cảm biến từ-địa bàn CMPS03[42]; b) Nguyên lý hoạt động.

Cảm biến cung cấp số liệu đo cường độ từ trường theo các trục hướng này, từ đó có thể xác định được góc lệch của mô-đun so với phương từ trường quả đất khi nó được gắn cố kết với tâm của robot. Một vi xử lý tích hợp trong mạch điện tử của mô-đun cung cấp các số đo này ra theo chuẩn truyền thông nối tiếp RS-232C. Cảm biến này có ưu điểm rất phổ biến và dễ sử dụng, có thể đo trực tiếp góc quay (hướng) của robot. Tuy nhiên vì lệ thuộc vào từ trường của trái đất nên nó có nhược điểm thường không chính xác khi hoạt động trong vùng mà từ trường cục bộ của trái đất bị ảnh hưởng do các vật nhiễm từ gần robot gây nên.

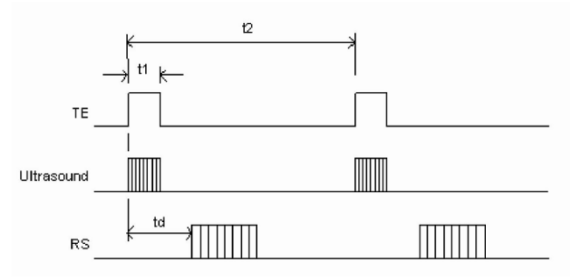
- *Cảm biến đo xa siêu âm (ultrasonic sensor).*

Cảm biến siêu âm cho các thông tin về khoảng cách đến vật cản với thời gian thu thập số liệu nhanh. Nguyên tắc hoạt động của nó được chỉ ra trong hình

2.6b. Một chùm sóng siêu âm được phát ra từ cảm biến và sau một thời gian nào đó thu được sóng phản xạ từ vật cản. Biết thời gian này và vận tốc sóng siêu âm truyền trong môi trường thì có thể xác định được khoảng cách từ cảm biến đến vật.



(a)



(b)

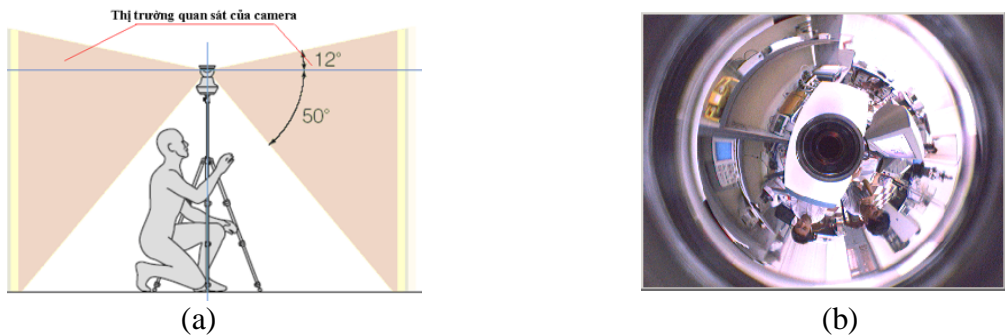
Hình 2.6 a) Mô-đun cảm biến siêu âm SRF05 [43]; b) Nguyên lý hoạt động.

Cảm biến siêu âm có ưu điểm là có giá thành thấp, dễ sử dụng và có lợi thế tạo ra kết quả nhanh hơn so với các thiết bị đo khác. Tuy nhiên, kết quả đo khoảng cách của cảm biến siêu âm bị một số hạn chế cơ bản do chùm tia siêu âm bị tán xạ thành hình nón chứ không phải là một tia hẹp. Độ chính xác của kết quả đo không ổn định do bị ảnh hưởng bởi các hiện tượng như: tiếng dội (echo), búp sóng bên (side lobe), phản xạ gương (specular reflection), xuyên âm (cross-talk) của chùm sóng siêu âm nên khó xác định được kích thước của vật. Vì những lý do trên, chúng tôi chỉ sử dụng các cảm biến này để phát hiện những vật cản gần, giữ an toàn cho robot. Tám mô-đun cảm biến siêu âm loại SRF05 đã được lắp đặt cho phép phát hiện được vật trong khoảng cách từ 3cm đến 4m với sai số là 2%; và góc mở chùm siêu âm phát ra khoảng $22,5^\circ$.

- *Cảm biến ảnh camera toàn phương (omni-directional camera).*

Cảm biến ảnh camera toàn phương được sử dụng là loại Hyper-Omni Vision SOIOS 55 như biểu diễn trên hình 2.7.

Nó gồm một gương lồi phản xạ hình ảnh trong toàn không gian cùng một camera thông thường được đặt ở mặt phẳng ảnh của gương như hình 2.7a. Ảnh toàn phương chứa đựng thông tin về vị trí của các đối tượng theo các góc hướng trong toàn không gian 360° như hình 2.7b [55].



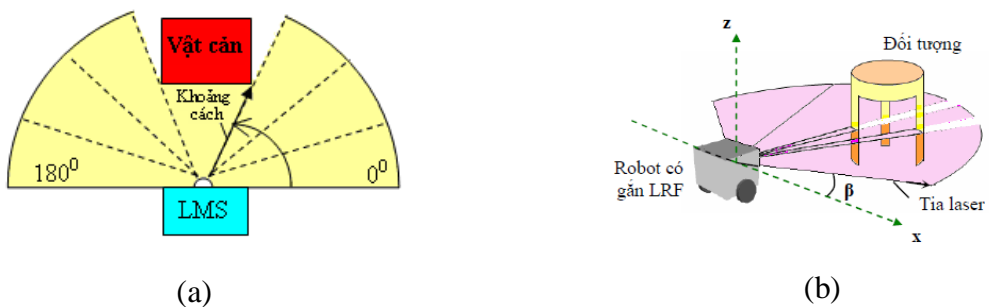
Hình 2.7 Cảm biến ảnh camera toàn phương. [55]

a) Thị trường quan sát của camera; b) Hình ảnh thu được của camera.

Camera này có ưu điểm cho phép quan sát đồng thời toàn không gian 360° quanh robot mà không phải quay liên tục camera. Ta có thể sử dụng camera toàn phương như một phép đo định hướng tuyệt đối và được tổng hợp với các phép đo khác để tạo ra một hệ thống thị giác hoàn chỉnh cho robot. Tuy nhiên cũng giống như các camera thông thường khác, các dữ liệu hình ảnh cũng bị ảnh hưởng nhiều vào điều kiện ánh sáng và môi trường hoạt động. Để camera làm việc tốt ta phải chọn một môi trường thích hợp, ví dụ làm nổi bật (thấp sáng) các cột mốc, cửa,... mà camera cần phát hiện.

- Cảm biến đo xa laser LRF (*laser range finder*).

Thiết bị đo xa laser LRF của hãng SICK với dải đo từ 0,4 m đến 80 m đã được lắp đặt trong robot. Đây là một thiết bị cho phép xác định nhanh chóng và chính xác khoảng cách và góc lệch từ robot tới vật cản.



Hình 2.8 a) Dải góc quét 180° của LRF; b) Mặt phẳng tia laser với các góc quét β .

Cảm biến hoạt động trên nguyên lý đo thời gian từ thời điểm một xung sáng laser hồng ngoại được phát ra từ LRF đến thời điểm thu được xung này khi nó được phản xạ từ vật cản trong môi trường. Một hệ thống đo thời gian cực kỳ chính xác

trong thiết bị cho tính khoảng cách đến vật khi biết vận tốc của ánh sáng trong môi trường. Bằng việc gắn một gương phản xạ quay trên đường đi của tia sáng, các xung laser này được làm chệch hướng khi gương quay. Gương phản xạ quay với tốc độ 75 vòng/giây tạo nên một dải quét tia laser hình rẻ quạt ở phía trước cảm biến với một trường nhìn 2D là 180° như hình 2.8a. Trong mỗi lần quét như vậy, các xung laser được phát đi chệch hướng với trục x một góc β với độ phân giải $0,25^\circ$; $0,5^\circ$ hoặc 1° như mô tả trên hình 2.8b. Lúc này, bộ phận điện tử trong LRF sẽ phát ra một tập số liệu về khoảng cách đến đối tượng và góc quét tương ứng ở lối ra [113]. Người sử dụng phải lập trình để thu thập chính xác tập số liệu cho một chu kỳ quay gương (chu kỳ quét tia laser), là 13,3 ms.

Cũng giống như các cảm biến thương phẩm khác, cần phải phát triển chương trình riêng cho thu thập và xử lý dữ liệu cho LRF 2D. Chương trình thu thập và xử lý dữ liệu từ LRF được chúng tôi phát triển trong môi trường Microsoft Visual C++.

Sau khi nhận lệnh yêu cầu truyền số liệu từ máy tính, LRF sẽ gửi về các khung dữ liệu đo tương ứng với mỗi mặt quét ngang. Định dạng dữ liệu khung gồm 7 byte tiêu đề, 2N byte số liệu với 2 byte cho một điểm đo và cuối cùng là 2 byte mã phát hiện lỗi CRC.

7 byte tiêu đề	2N byte số liệu	2 byte mã CRC
----------------	-----------------	---------------

Để định vị chính xác khối số liệu đo, chương trình cần phát hiện được 7 byte tiêu đề trong mỗi khung truyền liên tiếp. Muốn vậy, bộ đệm nhớ bên PC phải đủ rộng để chứa được hơn 2 lần số byte trong một khung truyền. Thí dụ, với chế độ dải quét ngang 100° , độ phân giải góc $0,25^\circ$; sẽ có 400 điểm đo ứng với 800 byte số liệu cộng với 7 byte tiêu đề và 2 byte CRC, thì bộ đệm có thể chọn cỡ từ 1.700 đến 2.000 byte. Kích thước này cũng không nên lớn quá vì làm chậm lại quá trình truyền dữ liệu. Lưu đồ của chương trình thu thập và xử lý dữ liệu được chỉ ra trên hình 2.9.

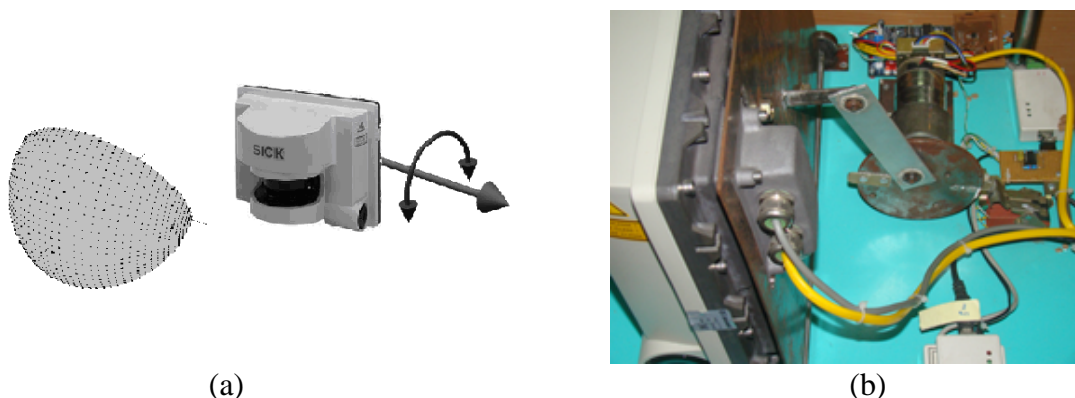


Hình 2.9 Lưu đồ chương trình thu và xử lý số liệu.

- Xây dựng cảm biến đo xa laser 3D từ thiết bị LRF-2D.

Thiết bị đo xa laser LRF là một cảm biến hiện đại, nó có nhiều ưu điểm như khả năng thu thập đo đạc khoảng cách với tốc độ và độ chính xác cao, kết quả không phụ thuộc nhiều vào điều kiện môi trường. Tuy vậy thông tin hình ảnh 2D đem lại có thể không đủ trong một số trường hợp cần phát hiện các vật có kết cấu không giống nhau theo chiều dọc (như bàn, các khung dầm ngang...) như trên hình 2.8b. Để khắc phục vấn đề này, có thể thay thế nó bằng một cảm biến laser 3D nhưng với giá thành lại rất cao. Cùng chung với xu thế trên thế giới, chúng tôi đã cải tiến LRF 2D thành một hệ đo xa 3D dựa trên thiết bị LRF 2D rẻ tiền hơn này. Hầu hết các giải pháp đều sử dụng một máy quét 2D (máy đo xa 2D) kết hợp với một cơ cấu cơ khí cho phép quay bộ máy theo chiều thứ ba, nghĩa là tạo nên một chiều quét thứ ba. Một vài phương pháp quét đã được sử dụng với các tên gọi là *pitching scan*, *rolling scan*, *yawing scan*, v.v.... Hệ thống của chúng tôi cũng được thiết kế là loại “pitching scan” nhưng cơ chế hoạt động của nó có khác so với các công trình đã nêu ra của các tác giả Wulf Oliver, et al. [125] và Harrison Alastair, et al. [56]. Trong thiết kế của các tác giả đó, LRF được gắn vào một đế được điều khiển cho quay

liên tục theo một chiều. Điều này cho phép có thể nhận được một tốc độ quay ổn định để đảm bảo cho một hình ảnh laser thu nhận có độ tuyến tính tốt theo chiều dọc. Tuy nhiên, nó lại dẫn đến việc phải thay thế các dây dẫn cấp điện và tín hiệu liên tục tới LRF bằng các vòng cổ góp cơ khí tiếp xúc điện. Chính loại tiếp xúc này sẽ sinh ra sự bất ổn định về tín hiệu và tăng can nhiễu của hệ thống. Ngược lại, trong hệ thống của chúng tôi, LRF được gắn lên một đế có thể quay ngẩng lên xuống quanh một trục nằm ngang, đế của LRF được thiết kế chỉ quay ngẩng lên - quay xuống rồi lặp lại với một dải góc nhất định nhỏ hơn 1 vòng xoay, và do đó không cần đến cổ góp điện. Trong quá trình quay ngẩng lên-xuống của LRF, mặt phẳng quét ngang tia laser sẽ được ngẩng lên-xuống theo, cho phép thu thập được một “đám mây” các điểm số liệu đo 3D được phân bố trong một phần không gian ảo hình cầu có bán kính là khoảng đo cực đại của LRF. Hình 2.10a cho hình ảnh của phương pháp quay này, quét ngẩng với trục quay lên và xuống cùng không gian hình cầu các điểm đo. Hình 2.10b cơ cấu truyền động quay và mô-tơ servo.



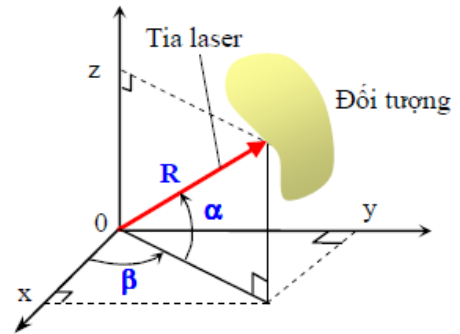
Hình 2.10 a) Quét ngẩng lên và xuống cùng không gian hình cầu các điểm đo; b) Cơ cấu truyền động quay và mô-tơ servo.

Dựa trên các số liệu này, ta có thể xác định được tọa độ Đề-các của mỗi điểm ảnh theo như hình 2.11. Mỗi tập số liệu của một mặt quét ngang (β , R) sẽ được kết hợp với một góc quét ngang α để tính các số liệu tọa độ các điểm ảnh. Sơ đồ thiết kế gắn kết LRF với các chi tiết cơ khí của hệ được thể hiện ở hình 2.12. Trong thời gian đo, hai tập giá trị về góc lệch ngang β của tia laser và khoảng cách đến vật R được truyền từ LRF vào máy tính.

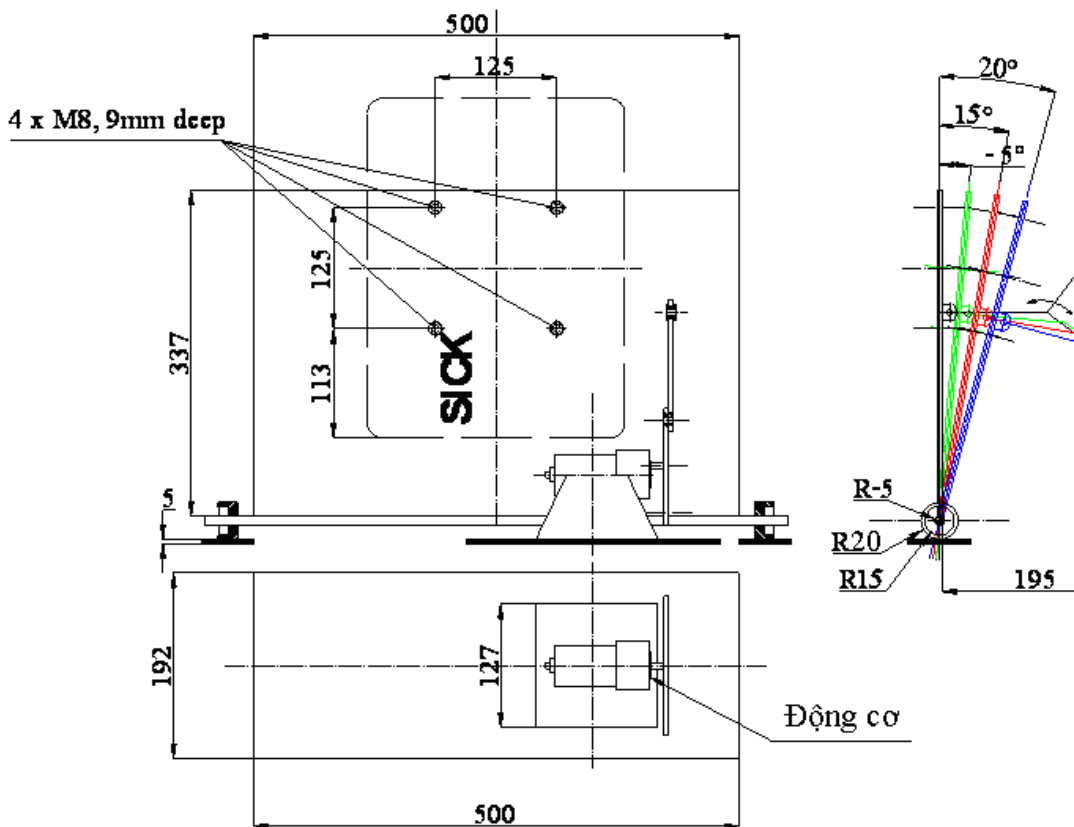
$$x = R \cos \alpha \cos \beta$$

$$y = R \cos \alpha \sin \beta$$

$$z = R \sin \alpha$$



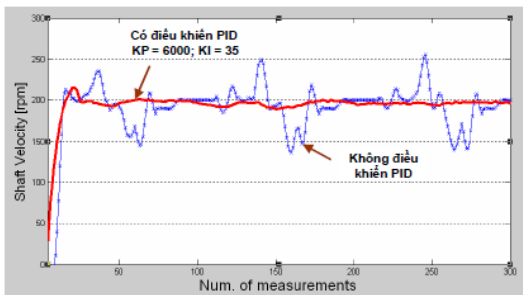
Hình 2.11 Xác định tọa độ của một điểm ảnh 3D.



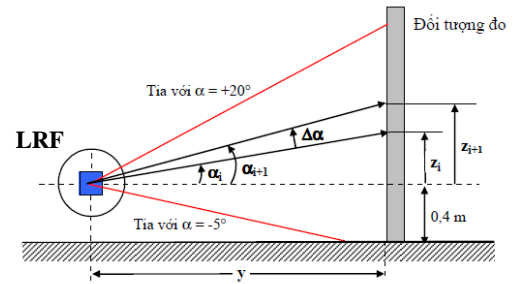
Hình 2.12 Chi tiết bản vẽ hệ thống cơ khí tạo quét ngẩng lên-xuống.

Để nhận được một hình ảnh chính xác theo chiều dọc, tốc độ quay ngẩng lên/xuống (tilt) của LRF cần được ổn định với một giá trị không đổi. Tuy nhiên, do sự bất đối xứng về cơ khí của hệ thống được thiết kế nên trong quá trình chuyển

động, các nhân tố như ma sát khớp nối, trọng lượng vật, v.v... sẽ gây nên sự bất ổn định của tốc độ quay. Nhằm khắc phục vấn đề này, một hệ thống điều khiển mô-tơ vòng kín servo đã được chúng tôi thực hiện (sử dụng mô-đun điều khiển mô-tơ *Motion Mind*). Tốc độ mô-tơ được đo bằng việc đếm số xung qua một bộ lập mã quang 500 khe sáng/vòng gắn với trục quay. Bằng phương pháp Ziegler–Nichols, các hệ số PID được đặt là $K_P = 6000$, $K_I = 35$ và $K_D = 2$. Sự ổn định của tốc độ mô-tơ quay ngừng đã được kiểm tra bởi một chương trình được viết trên ngôn ngữ hình tượng LABVIEW.



Hình 2.13 Tốc độ mô-tơ khi không và có điều khiển PID.



Hình 2.14 Góc quét dọc α của LRF.

Kết quả hình 2.13 chỉ ra rằng: khi không có điều khiển PID, tốc độ rất mất ổn định, đặc biệt tại các khoảng thời gian thay đổi hướng chuyển động. Ngược lại, khi có điều khiển PID, tốc độ được ổn định chỉ với biến đổi cỡ $\pm 5\%$.

Dải góc quét dọc α được thiết kế trong khoảng từ -5° đến $+20^\circ$ so với phương nằm ngang. Máy quét LRF được đặt trên khung robot cao so với mặt sàn là 0,4 m như chỉ ra trên hình 2.14. Việc nhận dữ liệu từ LRF để tạo một khung ảnh 3D được khởi phát từ thời điểm bắt đầu quét dọc (quay ngừng lên). Mạch điện với các rơ-le hành trình và mạch trigger thực hiện nhiệm vụ này. Gọi T_v là thời gian được đặt cho quá trình quét dọc của một khung ảnh 3D, đó là khoảng thời gian mô-tơ quay ngừng lên từ góc α bằng -5° đến $+20^\circ$. Việc đo góc α ứng với mỗi mặt quét ngang được thay bằng việc xác định số mặt quét n_h trong khoảng thời gian quét dọc 1 khung ảnh. Giá trị này phụ thuộc vào một số nhân tố như tốc độ truyền số liệu nối tiếp baud, độ dài số liệu tỷ lệ với số điểm đo trong từng chế độ quét ngang, thời

gian tìm kiếm các byte tiêu đề đánh dấu đồng bộ cho luồng số liệu, kích thước bộ đệm nhớ của LRF và máy tính. Vì vậy trong hệ thống thiết kế, n_h được xác định bằng thực nghiệm qua tổng số dòng số liệu quét ngang được chương trình thu thập trong thời gian quét dọc. Thực nghiệm cho thấy con số này là ổn định cho mỗi chế độ nên có thể suy ra bước nhảy $\Delta\alpha$ cho các mặt quét ngang liên kế là:

$$\Delta\alpha = \frac{5^\circ + 20^\circ}{n_h} \quad (2.1)$$

2.1.1.4. Khối điều khiển điện tử trong robot.

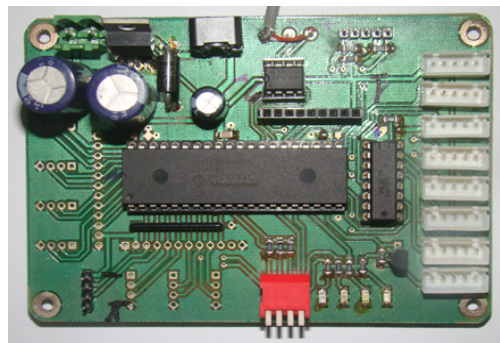
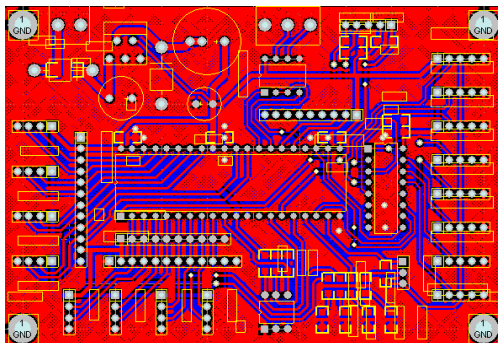
Ngoại trừ cảm biến ảnh toàn phương và cảm biến laser; các cảm biến còn lại được thông tin với máy vi tính qua vi điều khiển MCUdsPIC 30F4011, chương trình điều khiển được viết trên C++. Vi điều khiển MCUdsPIC giao tiếp với bên ngoài qua mạch chuyển đổi DS75176B. Đây là bo mạch cho phép chuyển đổi tín hiệu UART từ vi điều khiển ra chuẩn RS-485.

MCUdsPIC30F4011 là một dòng vi điều khiển 16 bit tích hợp bộ xử lý tín hiệu số 16 bit của hãng Microchip. Với việc tích hợp module xử lý tín hiệu số (Digital Signal Processor) và một lõi vi điều khiển (controller), hãng Microchip đã tạo ra một bộ vi điều khiển có khả năng xử lý các phép tính phức tạp và điều khiển trực tiếp các thiết bị ngoại vi chỉ với một linh kiện duy nhất. Không chỉ làm gọn nhỏ các bo mạch, một chip như vậy còn làm đơn giản việc thiết kế mạch điện tử và đặc biệt là tăng khả năng chống nhiễu cho hệ thống. Sơ đồ nguyên lý của mạch điện tử điều khiển dùng MCUdsPIC30F4011 được thiết kế như hình 2.16 và hình 2.15 là mạch in và ảnh chụp mạch lắp ráp của vi điều khiển MCUdsPIC.

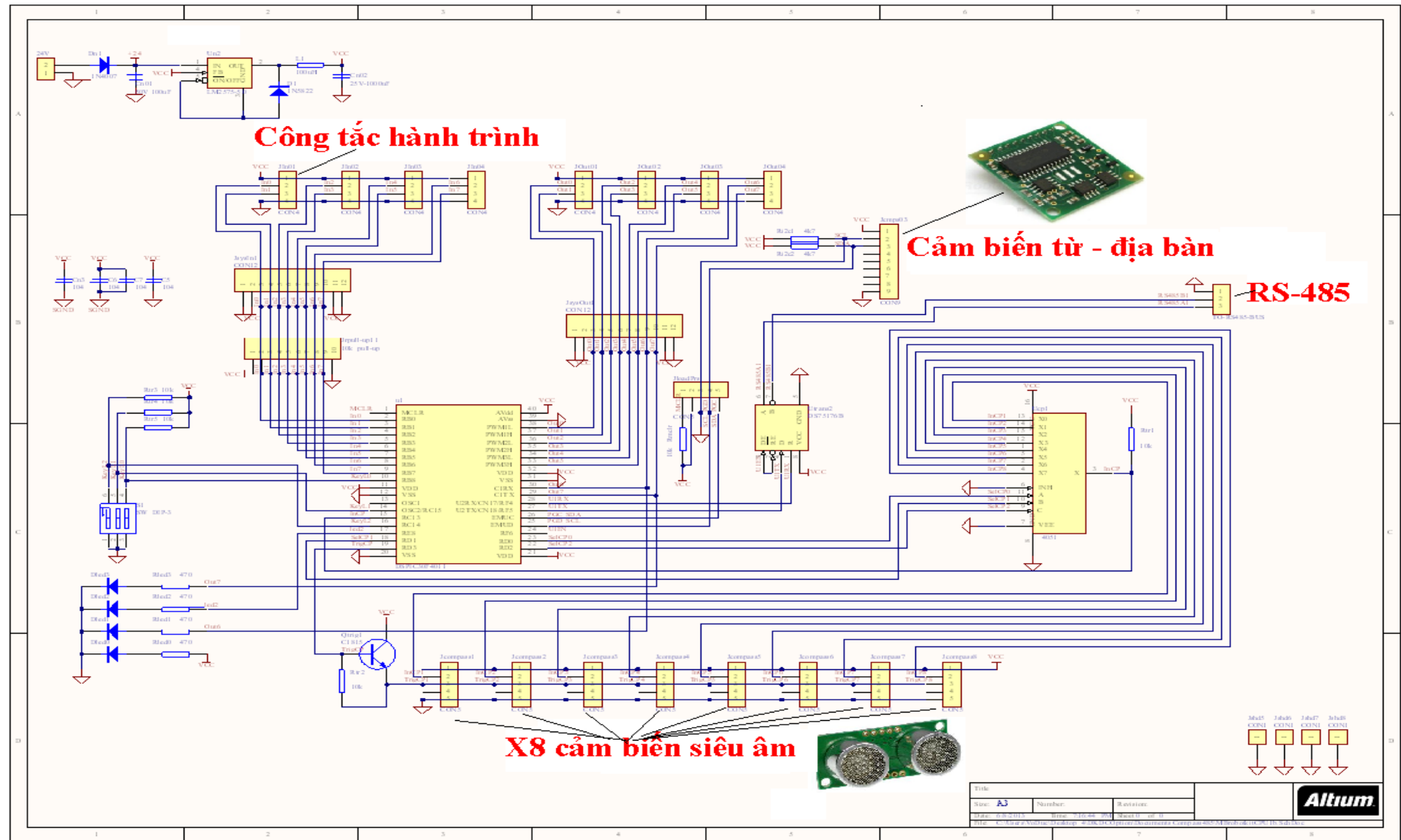
- *Các tính năng chính của dsPIC30F401.*

- + Kiến trúc Harvard có hiệu chỉnh;
- + Tập lệnh gồm 84 lệnh cơ bản với độ dài lệnh 24 bit, độ dài dữ liệu 16 bit;
- + 48 kB bộ nhớ Flash, có thể nạp lại nhiều lần, 2 kB bộ nhớ RAM;
- + 1kB bộ nhớ EEPROM cho phép lưu không bay hơi các tham số ;
- + Tốc độ xử lý tối đa 30 triệu lệnh trong 1 giây (MIPs);
- + 30 nguồn tạo ngắt với 3 nguồn ngoài, 8 mức ưu tiên;

- + Bộ xử lý tín hiệu số được tích hợp.
- *Các thiết bị ngoại vi trên chip.*
- + Mô-đun định thời (timer) với tần số xung clock có thể chia được và cấu hình thành 1 cặp timer 16 bit hoặc 1 timer 32 bit;
- + Bộ điều chế độ rộng xung (PWM) với độ phân giải 16 bit;
- + Mô-đun truyền dữ liệu MPI hỗ trợ chế độ nhiều chủ tớ;
- + 2 mô-đun truyền thông nối tiếp (UART) với bộ đệm FIFO, mô-đun truyền thông CAN tương thích chuẩn 2.0B;
- + 4 bộ chuyển đổi tín hiệu tương tự - số (ADC) với các khả năng: Tốc độ lấy mẫu lên tới 500Ksps; Chuyển đổi ngay cả trong chế độ ngủ (sleep).
- *Các tính năng khác của vi điều khiển.*
- + Bộ nhớ chương trình Flash với khả năng ghi/xóa tới 100.000 lần;
- + Bộ nhớ EEPROM với khả năng ghi/xóa 1 triệu lần; Có khả năng tự lập trình lại bằng mã lệnh; chế tạo theo công nghệ CMOS, hoạt động với dải điện áp rộng 2,5-5,5V; dải nhiệt độ công nghiệp và dải nhiệt độ mở rộng, tiêu thụ năng lượng thấp.



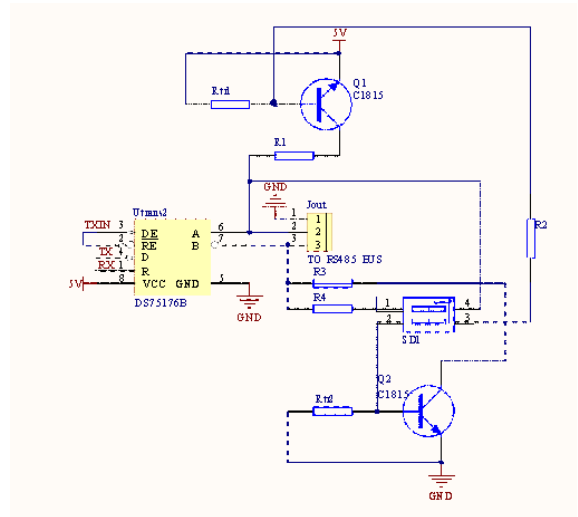
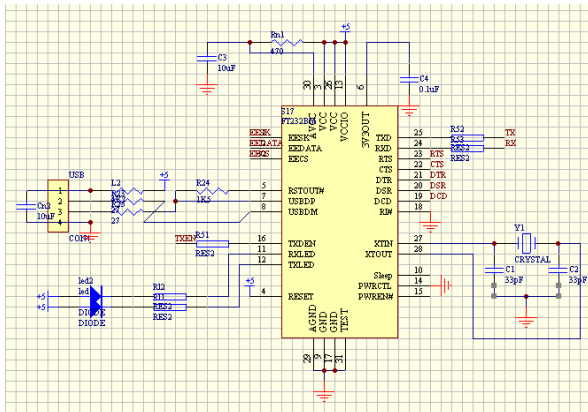
Hình 2.15 Mạch in và ảnh chụp mạch lắp ráp vi điều khiển MCUdsPIC.



2.1.1.5. Các khối giao tiếp truyền tin giữa vi điều khiển MCUsPIC với máy tính.

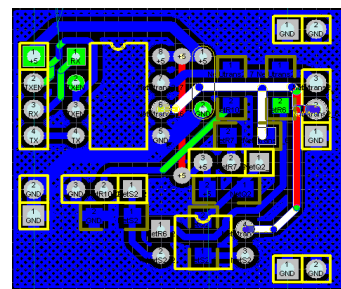
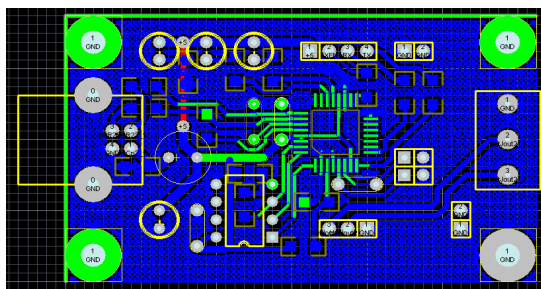
Khối giao tiếp truyền tin giữa vi điều khiển và vi tính được thiết kế thành bộ chuyển đổi RS-485 sang USB và ngược lại. Sơ đồ nguyên lý và sơ đồ lắp ráp đưa ra trên hình 2.17, 2.18 và 2.19.

- Sơ đồ nguyên lí.



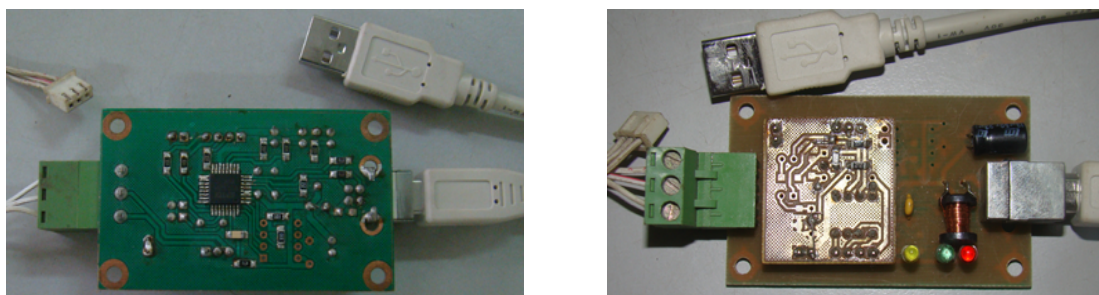
Hình 2.17 Sơ đồ nguyên lí của mạch chuyển đổi USB ↔ RS485.

- Sơ đồ mạch in.



Hình 2.18 Sơ đồ mạch in của mạch chuyển đổi USB ↔ RS485.

- Ảnh chụp mạch lắp ráp.



Hình 2.19 Ảnh chụp mạch lắp ráp mạch chuyển đổi USB ↔ RS485.

2.1.2. Chương trình điều khiển hệ thống.

Xuất phát từ cấu hình phần cứng trình bày ở trên, có thể chia phần mềm thu thập thông tin từ các cảm biến thành các mô-đun có các đặc điểm riêng sau.

- Chương trình thu thập và xử lý dữ liệu từ máy đo xa laser LRF là loại đặc biệt được phát triển riêng trong ngôn ngữ Visual C++;

- Thông tin hình ảnh toàn phương được thu thập trực tiếp vào máy vi tính và được xử lý qua chương trình được phát triển trên môi trường Visual C++ với công cụ là gói phần mềm mở xử lý ảnh OpenCV của Intel;

- Chương trình điều khiển cho 3 mạch đo bộ lập mã quang (được gắn trên 2 bánh xe chủ động và 1 mô-tơ quay cảm biến laser LRF) được kết nối trực tiếp vào máy vi tính được truyền qua đường truyền RS-232C;

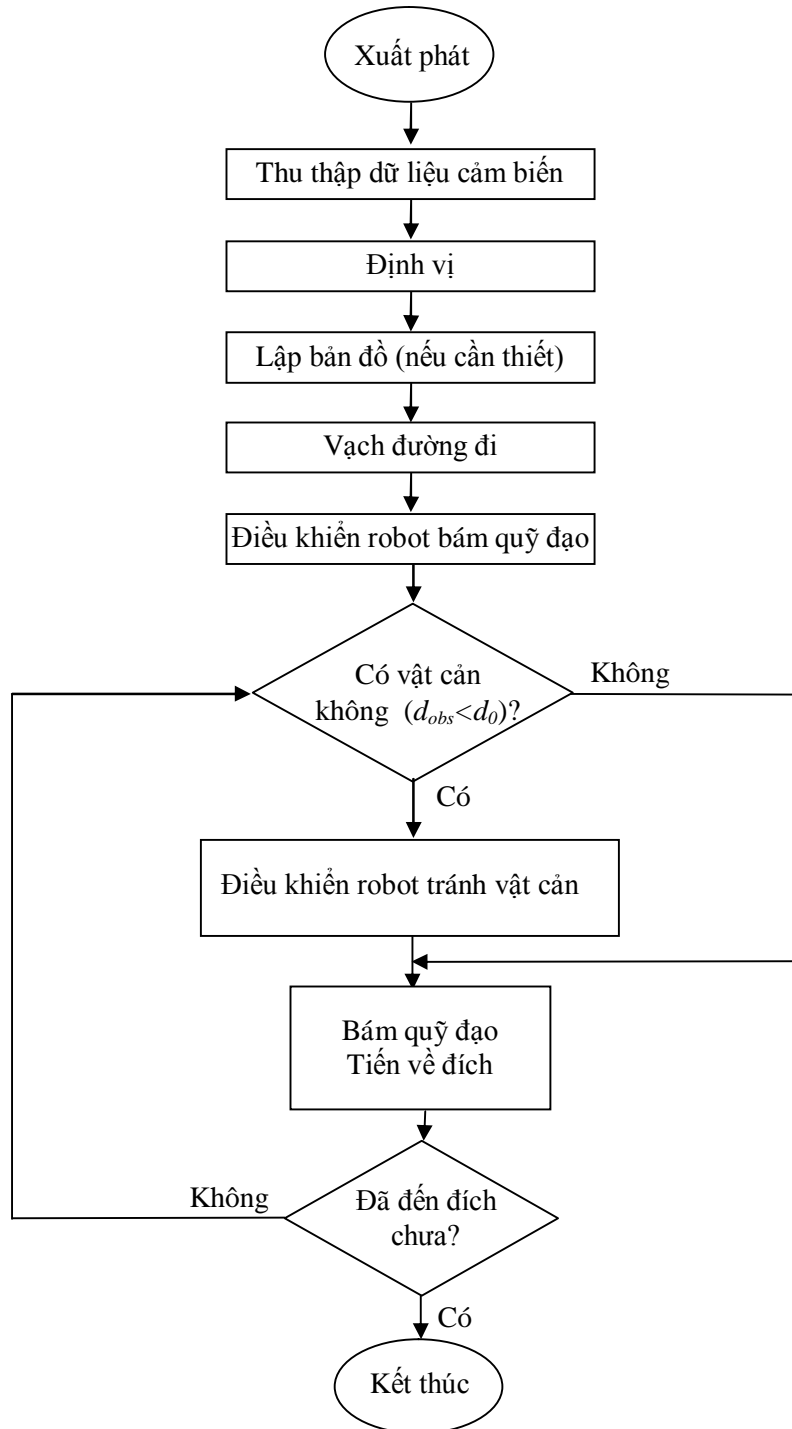
- Các thông tin còn lại được trao đổi từ 10 nút mạng với nút điều khiển từ máy tính (1 nút cho cảm biến từ-địa bàn chỉ hướng, 1 nút cho công tắc hành trình, 8 nút cho 8 cảm biến siêu âm) được truyền qua đường truyền RS-485 sử dụng giao thức Modbus/RTU.

- Trên hình 2.20 là lưu đồ khái quát quá trình dẫn đường cho robot di động, ta có thể tóm lược như sau:

1. Đầu tiên robot thu thập dữ liệu qua các cảm biến.
2. Xác định vị trí của robot.
3. Lập bản đồ (nếu cần thiết).
4. Vạch quỹ đạo.

5. Điều khiển robot chuyển động bám quỹ đạo đến đích, tránh vật cản.

Chi tiết thực thi các bước sẽ được trình bày chi tiết ở 2 chương tiếp sau của Luận án.

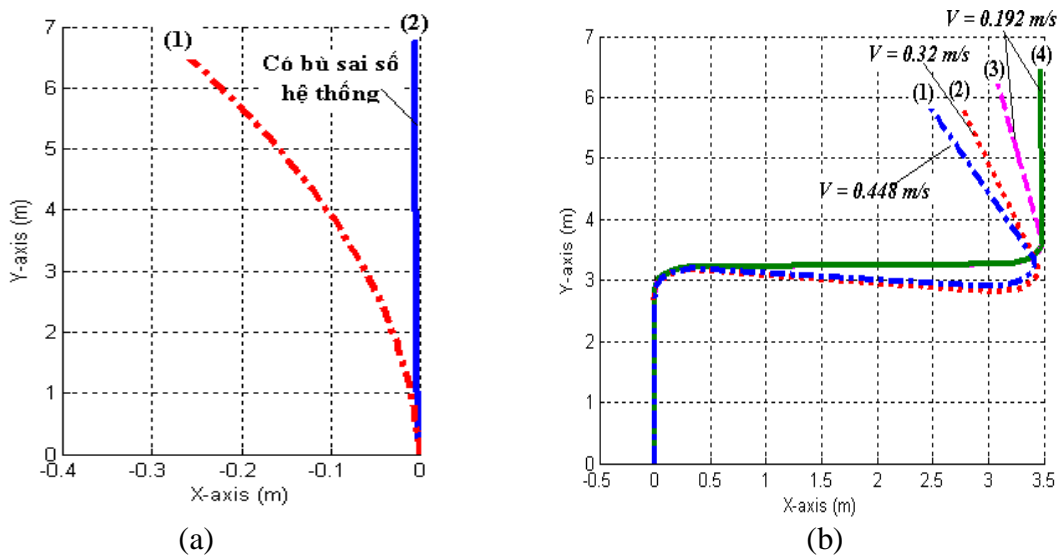


Hình 2.20 Lưu đồ quy trình dẫn đường cho robot di động.

2.2. Đo đặc đánh giá mô hình hệ thống được chế tạo.

2.2.1. Kiểm tra độ chính xác của chuyển động robot.

Trên cơ sở robot đã được thiết kế chế tạo chúng tôi đã tiến hành khảo sát độ chính xác chuyển động của robot. Cho robot tiến về phía trước, quay trái, quay phải trong 10 lần với các vận tốc khác nhau (0,19m/s, 0,32m/s và 0,45m/s). Các kết quả đo quỹ đạo thực của robot được thể hiện ở hình 2.21a và 2.21b.



Hình 2.21 a) Robot chạy thẳng 7m; b) Robot chạy và quay với các vận tốc khác nhau.

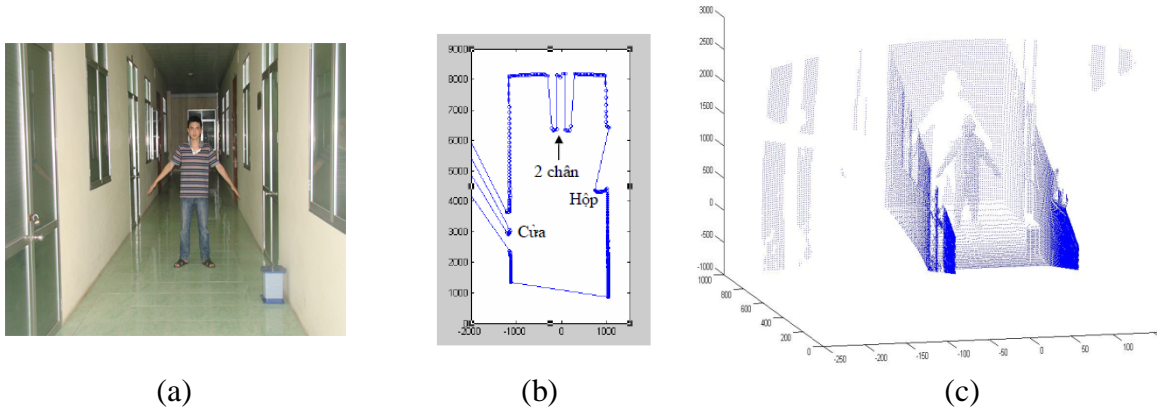
Trên hình 2.21a, khi điều khiển robot chạy thẳng tiến về trước, do có sai số hệ thống là 3,86%, nên quỹ đạo thực của robot (1) bị lệch theo thời gian so với đường mong muốn. Áp dụng phương pháp bù sai số hệ thống của J. Borenstein [22], chất lượng được cải thiện như quỹ đạo (2) với sai số là 0,09%.

Hình 2.21b biểu diễn kết quả đo quỹ đạo thu được trên các thử nghiệm: đi thẳng 3m \rightarrow quay phải 90° \rightarrow đi thẳng 3m \rightarrow quay trái 90° \rightarrow đi thẳng 3m và dừng. Với các vận tốc khác nhau 0,19m/s; 0,32m/s; 0,45m/s thu được các quỹ đạo thực của robot tương ứng với các đường cong (1), (2) và (3). Áp dụng phương pháp bù sai số hệ thống của J. Borenstein ta được quỹ đạo (4)

Chính những khảo sát ban đầu này sẽ làm cơ sở để tính toán các ma trận hiệp phương sai của phép đo bằng phương pháp Odometry sẽ được đề cập tới ở phần sau.

2.2.2. Kiểm tra độ tin cậy của ảnh laser.

Tiếp đến tiến hành đo đạc thực nghiệm thu thập và xử lý ảnh laser 3D đã được tiến hành trong phòng với môi trường tĩnh và động.

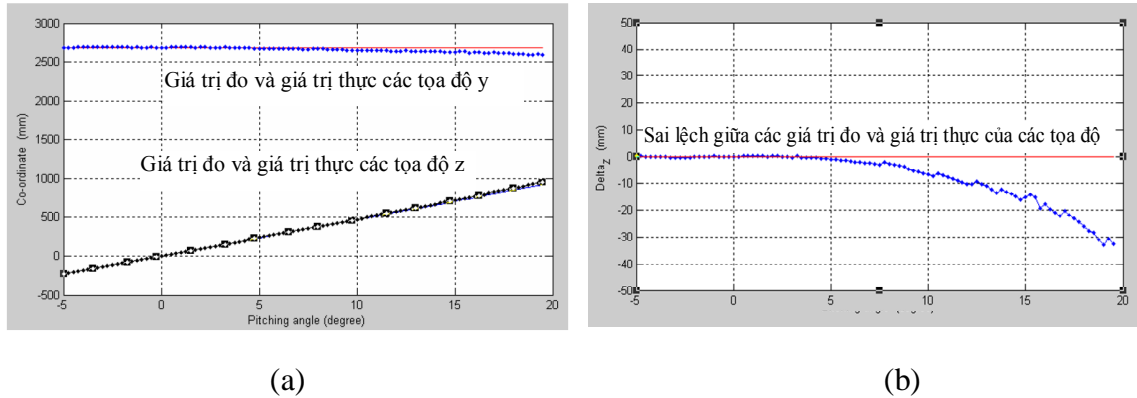


Hình 2.22 a) Ảnh camera; b) Ảnh laser 2D; c) Ảnh laser 3D.

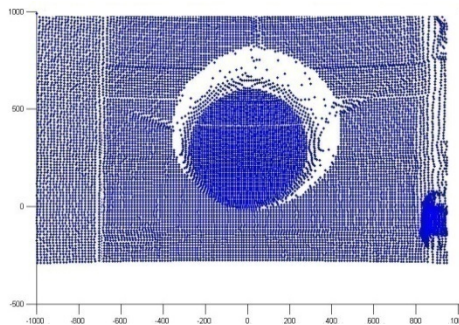
- Hình 2.22a là ảnh camera một người đứng yên trong hành lang dùng để so sánh. Hình 2.22b là ảnh laser 2D với góc quét ngang của LRF là 0° (mặt phẳng quét ngang song song với mặt sàn). Do LRF được đặt trên robot cao hơn mặt sàn 0,4 m nên chỉ có 2 chân người là được phát hiện. Trong khi đó, ảnh laser 3D trên hình 2.22c cho ta bức tranh của toàn bộ cơ thể người theo chiều cao.

- Như trên đã nói, do tốc độ quay mô tơ servo ổn định nên các giá trị góc ngang α trong hệ thống không cần phải đo trực tiếp mà được áp đặt ngay theo công thức (2.1). Vì vậy việc khảo sát sự sai khác của các số liệu tọa độ thu được trên cơ sở đo đạc và số liệu thực là cần thiết vì nó cho biết độ chính xác và tuyến tính của hình ảnh. Quá trình khảo sát được thực hiện trên một đối tượng là một vật phẳng được đặt chính xác theo phương thẳng đứng, cách cảm biến một khoảng đã biết là 2.850 mm giống như mô tả trên hình 2.22. Tại tọa độ $x = 0$, các kết quả đo đạc so sánh giữa giá trị đo và giá trị thực của sự phụ thuộc của các tọa độ y (khoảng cách đến vật) và z (chiều cao của vật) vào các góc ngang được chỉ ra trên các hình 2.23a và 2.23b. Sai số theo trục y và z đều cho thấy không quá 3%. Sai số này nằm trong khoảng sai số đo của thiết bị LRF. Hình 2.24 là mặt cắt (z,x) của hình 3D một vật hình tròn đặt cách LRF một khoảng 2,59 m và bức tường cách 2,90 m.

Kết quả trực quan cho thấy độ sai lệch là nhỏ so với các kích thước đo. Do đó có thể kết luận là hệ thống servo hoàn toàn đảm bảo cho một hình ảnh 3D có độ tin cậy chấp nhận được cho ứng dụng robot.



Hình 2.23 a) Các cặp giá trị đo (đường đứt nét) tọa độ y và z của vật và giá trị thực (đường liền nét) phụ thuộc vào góc nghiêng; b) Sai lệch tuyệt đối của các giá trị z.



Hình 2.24 Ảnh 3D một vật hình tròn đặt trước LRF.

- Hệ hoạt động với các thông số ứng với 2 thời gian quét dọc T_v ngắn và dài được cho trên bảng sau.

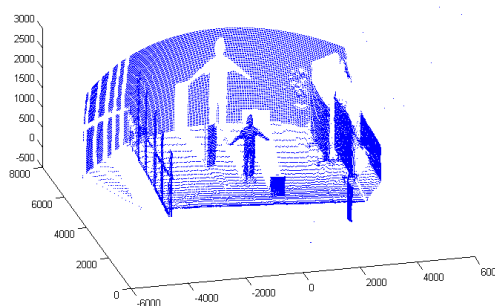
Bảng 2.1 Các thông số ứng với 2 thời gian quét dọc T_v ngắn và dài.

T_v	Phân giải góc $\Delta\beta$	Số mật quét ngang /1 khung	Số mật quét ngang / 1 giây
10 s	1.0°	81	8.1
	0.5°	40	4.0
	0.25°	24	2.4
42 s	1.0°	320	7.6
	0.5°	160	3.8
	0.25°	100	2.4

Việc chọn thời gian quét dọc tùy thuộc vào từng ứng dụng. Thời gian quét chậm với độ phân giải cao, thí dụ với ($T_v = 42$ s, $\Delta\beta = 0,25^\circ$), được dùng cho lập bản đồ toàn phòng khi robot đứng yên. Robot sẽ dừng ở một vị trí và thu thập số liệu trong 42 s để xây dựng bản đồ toàn phòng. Hình 2.25a là ảnh camera để so sánh. Hình 2.25b là ảnh laser 3D thu được. Kết quả cho thấy độ chính xác, độ phân giải và tuyến tính là chấp nhận được.

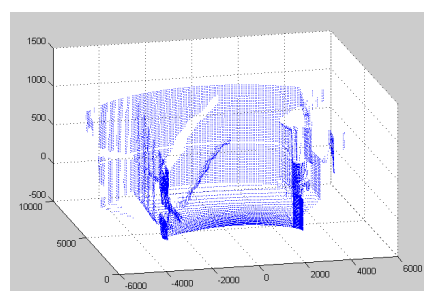
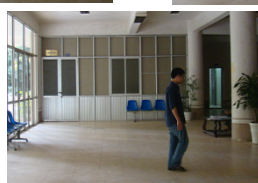


(a)

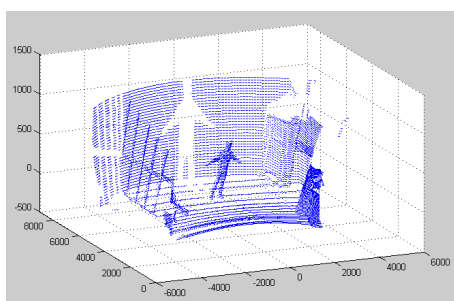


(b)

Hình 2.25 a) Ảnh camera; b) Ảnh Laser 3D.



Hình 2.26 Ảnh 3D một người đi ngang qua LRF.



Hình 2.27 Ảnh 3D một người đứng tại chỗ trong khi robot di chuyển về phía đó với vận tốc 0,3 m/s.

Để áp dụng cho môi trường động cần thời gian quét nhanh, ví dụ với trường hợp $T_v = 10$ s, $\Delta\beta = 1^\circ$. Đó là trường hợp dùng cho robot di động tránh vật cản và lập bản đồ cục bộ. Hình 2.26 là kết quả bắt ảnh laser 3D một người đang di chuyển ngang qua LRF trong khi robot đứng yên và hình 2.27 là trường hợp người đứng tại chỗ trong khi robot di chuyển về phía trước với vận tốc 0,3 m/s.

2.3. Kết luận.

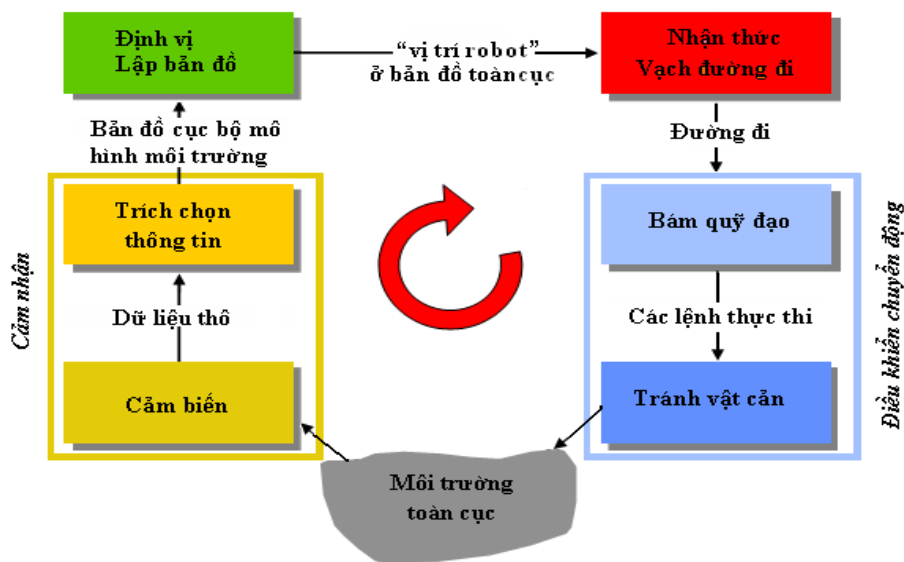
Trong chương này tác giả đã trình bày việc thiết kế chế tạo cả về phần cứng lẫn phần mềm điều khiển chung một mô hình robot di động đa cảm biến. Đặc biệt một hệ đo xa laser 3D dựa trên máy đo 2D đã được xây dựng thành công. Các kết quả đo đạc thực nghiệm cho thấy hệ thống hoàn toàn sử dụng được cho các nhiệm vụ định vị và lập bản đồ của robot trong phòng thí nghiệm. Mô hình này sẽ được sử dụng làm nền tảng để phát triển một số phương pháp dẫn đường dựa trên tổng hợp các cảm biến cho robot di động sẽ được trình bày sau. Các kết quả đã công bố tại Danh mục các công trình khoa học của tác giả liên quan đến Luận án [1-2].

CHƯƠNG 3

TỔNG HỢP CẢM BIẾN DÙNG CHO ĐỊNH VỊ VÀ LẬP BẢN ĐỒ

DẪN ĐƯỜNG ROBOT DI ĐỘNG

Mục đích thiết kế một robot di động là làm cho nó có khả năng xác định chính xác vị trí của mình trong môi trường, sau đó được điều khiển đi đến đích an toàn theo một quỹ đạo mong muốn. Quá trình đó gọi là dẫn đường robot di động. Tùy thuộc vào ứng dụng mà robot có thể được thiết kế hoạt động tự trị, bán tự trị, hoặc điều khiển từ xa bằng tay. Nhìn chung quá trình dẫn đường có thể bao gồm 4 khâu: cảm nhận, định vị-lập bản đồ, vạch đường đi và điều khiển chuyển động theo chu trình như thể hiện trên hình 3.1.

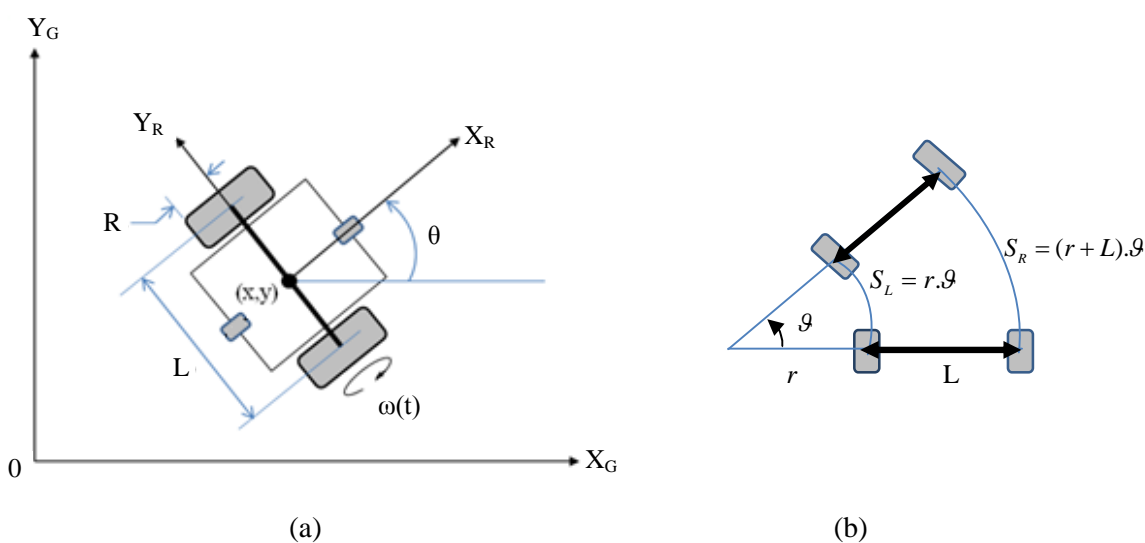


Hình 3.1 Các khâu điều khiển trong quá trình dẫn đường [111].

Khâu định vị-lập bản đồ sử dụng cơ sở dữ liệu từ khâu cảm nhận để làm khớp với dữ liệu bản đồ đã biết trước hoặc được lập từ cơ sở dữ liệu đó. Như đã thảo luận ở chương 1, kết quả của việc tổng hợp cảm biến có thể áp dụng đến tất cả các khâu kể trên, trong đó khâu định vị là rất quan trọng. Trong chương này chúng tôi sẽ giới thiệu việc áp dụng phương pháp tổng hợp cảm biến để nâng cao độ tin cậy của phép định vị robot và việc thực nghiệm một cách thức xây dựng bản đồ dẫn đường khi sử dụng phương pháp này.

3.1. Tổng hợp cảm biến bằng bộ lọc Kalman mở rộng để nâng cao độ tin cậy của phép định vị robot.

Định vị là việc xác định tọa độ và hướng của robot trong môi trường. Hình 3.2. biểu diễn hệ tọa độ và ký hiệu các thông số của robot di động được thiết kế trong luận án có thể di chuyển trên mặt sàn phẳng trong phòng, trong đó (X_G, Y_G) là các trục hệ tọa độ toàn cục của phòng, (X_R, Y_R) là các trục hệ tọa độ cục bộ gắn với tâm robot.



Hình 3.2 a) Tư thế và các thông số của robot trong hai hệ tọa độ; b) Mô hình chuyển động.

Robot có hai *bánh xe chủ động* được gắn trên một trục và hai *bánh xe bị động* được đặt ở phía trước và phía sau để cân bằng để robot. Hai bánh bị động này được thiết kế đủ linh động để không ảnh hưởng gì đến trạng thái động học của robot. Kết quả là trạng thái chuyển động của robot chỉ phụ thuộc vào 2 bánh xe chủ động. Các bánh chủ động có cùng bán kính R và cách nhau một khoảng L . Chúng được điều khiển độc lập bằng các mô tơ điện cho phép robot chuyển động theo hướng bất kỳ (nếu tốc độ quay 2 bánh bằng nhau thì robot sẽ đi thẳng, nếu tốc độ quay khác nhau thì robot sẽ quay hướng về phía bánh xe có tốc độ thấp hơn). Vì lý do trên nên loại robot kiểu này thường được gọi là robot có bánh xe được điều khiển vi sai (differential drive wheeled robot).

Bài toán được đặt điều kiện robot là một vật rắn chuyển động trên mặt sàn phẳng. Tâm của robot (trọng tâm) nằm tại điểm giữa của trục hai bánh xe chủ động.

Tư thế của robot (tọa độ và hướng) trong hệ tọa độ toàn cục được biểu diễn bởi véc-tơ trạng thái trạng thái $\mathbf{x} = [x, y, \theta]^T$, trong đó x, y là tọa độ tâm robot và θ là góc hướng của robot theo phương trục X_R so với trục tọa độ phòng X_G .

Khi đặt vận tốc góc của hai bánh xe phải và trái tương ứng là ω_R và ω_L thì với khoảng thời gian lấy mẫu Δt đủ ngắn ta có thể có các tín hiệu điều khiển đầu vào là số gia quãng đường dịch chuyển đặt lên các bánh xe này là:

$$\Delta s_R = \Delta t \cdot R \cdot \omega_R \quad \text{và} \quad \Delta s_L = \Delta t \cdot R \cdot \omega_L$$

Từ đây có số gia quãng đường dịch chuyển của tâm robot Δs là:

$$\Delta s = \frac{\Delta s_R + \Delta s_L}{2}$$

và số gia góc hướng mà robot quay được $\Delta \theta$ là:

$$\Delta \theta = \frac{\Delta s_R - \Delta s_L}{L}$$

Vậy phương trình trạng thái của robot tại thời điểm k trong hệ tọa độ toàn cục được cập nhật từ thời điểm $k-1$ như sau:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta s_k \cos(\theta_{k-1} + \Delta \theta_k / 2) \\ \Delta s_k \sin(\theta_{k-1} + \Delta \theta_k / 2) \\ \Delta \theta_k \end{bmatrix} \quad (3.1a)$$

Các số gia dịch chuyển Δs_R và Δs_L (tỷ lệ với tốc độ đặt của 2 bánh xe ω_R và ω_L và được đo bởi các cảm biến lập mã trục quay) là các biến đầu vào chịu ảnh hưởng của các sai số hệ thống và sai số ngẫu nhiên.

Các sai số hệ thống thường được sinh ra bởi sự không hoàn hảo về thiết kế và chế tạo cơ khí. Ngoài sai số độ phân giải của cảm biến lập mã trục quay, tốc độ lấy mẫu tín hiệu không đồng nhất, v.v...; hai nguồn sai số hệ thống điển hình nhất là số đo bán kính R của các bánh xe không đồng đều và sự không xác định độ dài trục L . Độ dài trục xe, nói chính xác chính là khoảng cách giữa hai điểm chạm của các bánh xe với mặt sàn, thực ra còn chịu sai số bởi các nhân tố ngẫu nhiên khác như sự gồ ghề của mặt sàn, lốp bánh xe không đủ cứng, v.v... Các sai số này có thể

bù trừ được bằng sự hiệu chuẩn cẩn thận (calibration) như các tác giả [22] đã làm. Trong luận án này, chúng tôi đã coi R và L là hằng số và các sai số hệ thống đã được bù trừ như kết quả được trình bày trong 2.2.1.

Các sai số ngẫu nhiên được sinh ra bởi các tương tác không đoán trước được của robot với môi trường như sàn nhà không đủ phẳng, các bánh xe bị trượt khi lăn trên sàn, lực cản do các bánh xe bị động gây nên, v.v... Các sai số này là trở ngại chính cho các ứng dụng robot di động và khó được bù trừ bằng các phương pháp thông thường.

Trong luận án, các sai số được quy về là tỉ lệ với các số gia dịch chuyển hai bánh xe Δs_R và Δs_L . Trong phương trình hệ thống, các số gia này khi chịu tác động của nhiễu có thể được biểu diễn thành tổng của hai thành phần gồm giá trị danh định và can nhiễu:

$$\begin{aligned}\Delta s_R &= \Delta s_{R0} + \varepsilon_R \\ \Delta s_L &= \Delta s_{L0} + \varepsilon_L\end{aligned}$$

trong đó, Δs_{R0} và Δs_{L0} là các giá trị danh định của tín hiệu đầu vào.

ε_R và ε_L là các giá trị nhiễu quá trình (process noise), được giả định là nhiễu trắng, độc lập, có trị trung bình bằng không và phân bố chuẩn.

Đến đây, phương trình trạng thái ở trên có thể được viết chi tiết hơn như sau:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) + (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2} \cos \left(\theta_{k-1} + \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2L} \right) \\ \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) + (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2} \sin \left(\theta_{k-1} + \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2L} \right) \\ \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{L} \end{bmatrix} \quad (3.1b)$$

Các công thức trên thuộc về phương pháp có tên là *Odometry* cho thấy xử lý này cũng thuộc về loại định vị *Dead-reckoning* [22,50], trong đó vị trí hiện tại của robot (\mathbf{x}_k) được xác định dựa trên vị trí trước đó của nó (\mathbf{x}_{k-1}). Ngoài ưu điểm là đơn giản với kết cấu hệ thống dễ thực hiện, phương pháp định vị này gặp phải đặc tính tích lũy của các sai số trong hệ thống, nên độ chính xác của việc ước lượng vị trí sẽ càng bị kém đi khi robot di chuyển càng dài. Có nhiều công trình nghiên cứu cố gắng tăng độ tin cậy của việc định vị. Cũng như xu hướng chung, một số công trình

của chúng tôi gần đây cũng đã áp dụng phương pháp tổng hợp cảm biến bằng thuật toán lọc Kalman để định vị robot được chế tạo và đã nhận được kết quả các phép đo vị trí có chất lượng cải thiện đáng kể.

Như đã đề cập trong chương 1, có thể áp dụng bộ lọc Kalman mở rộng EKF để giải quyết bài toán ước lượng véc-tơ trạng thái $\mathbf{x} \in \mathfrak{R}^n$ của một quá trình điều khiển được biểu diễn bởi phương trình (3.1b). Phương trình phi tuyến này có dạng (1.11) như đã nói trong chương 1:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \quad (3.2)$$

trong đó f là hàm hệ thống với $\mathbf{x} = [x, y, \theta]^T$ là véc-tơ trạng thái, $\mathbf{u} = [\Delta s_R, \Delta s_L]^T$ là véc-tơ điều khiển đầu vào và $\mathbf{w} = [\varepsilon_R \ \varepsilon_L]$ là nhiễu quá trình.

Trạng thái này được quan sát bởi một số phép đo tạo nên véc-tơ lỗi ra \mathbf{z} :

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (3.3)$$

trong đó h là hàm đo và $\mathbf{v} = [\varepsilon_{Rm} \ \varepsilon_{Lm}]$ là *nhiều đo* ảnh hưởng lên phép đọc các số gia dịch chuyển từ các bộ lập mã trực quay. Chỉ số m ở đây để phân biệt cho nhiều đo.

\mathbf{w}_k và \mathbf{v}_k được cho là độc lập với nhau, là ồn trắng, có phân bố xác suất chuẩn với các ma trận hiệp phương sai tương ứng là Q và R :

$$P(w) \sim N(0, Q) \quad P(v) \sim N(0, R) \quad E(\mathbf{w}_i \mathbf{v}_j^T) = 0$$

Dựa trên các số liệu đo \mathbf{z}_k , bộ lọc Kalman tìm được một hệ số lọc K_k trong mỗi chu kỳ lấy mẫu tín hiệu đo thuộc một vòng lặp đệ quy, sao cho ước lượng trạng thái của hệ gần với giá trị thực nhất. Một lần nữa, các phương trình tính EKF áp dụng cho các hệ phi tuyến được thực hiện qua các pha sau:

Khởi đầu: EKF được khởi phát với ước lượng trạng thái hậu nghiệm $\hat{\mathbf{x}}_0$ và hiệp phương sai của ước lượng sai số P_0 ở bước thời gian 0.

1. **Pha dự đoán** (prediction): ở mỗi bước thời gian k , bộ lọc truyền trạng thái \mathbf{x} và hiệp phương sai P của hệ thống ở bước trước tới bước hiện tại khi sử dụng các phương trình cập nhật thời gian:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0) \quad (3.4)$$

$$P_k^- = A_k P_{k-1}^- A_k^T + W_k Q_{k-1} W_k^T \quad (3.5)$$

ở đây,

- A_k là ma trận Jacobian có thành phần A_{ij} là các đạo hàm riêng của hàm f đối với trạng thái x :

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0).$$

- W_k là ma trận Jacobian có thành phần W_{ij} là các đạo hàm riêng của hàm f đối với ồn w :

$$W_{ij} = \frac{\partial f_i}{\partial w_j}(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, 0).$$

2. **Pha hiệu chỉnh** (Correction): Bộ lọc sửa ước lượng trạng thái tiên nghiệm với các phép đo \mathbf{z}_k bởi các phương trình cập nhật số đo sau:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (3.6)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + K_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-, 0)) \quad (3.7)$$

$$P_k = (I - K_k H_k) P_k^- \quad (3.8)$$

ở đây,

- I là ma trận đơn vị,
- H_k là ma trận Jacobian có thành phần H_{ij} là các đạo hàm riêng của hàm đo h đối với trạng thái x :

$$H_{ij} = \frac{\partial h_i}{\partial x_j}(\hat{\mathbf{x}}_k, 0).$$

- V_k là ma trận Jacobian có thành phần V_{ij} là các đạo hàm riêng của hàm đo h đối với ồn v :

$$V_{ij} = \frac{\partial h_i}{\partial v_j}(\hat{\mathbf{x}}_k, 0).$$

Để tổng hợp dữ liệu cảm biến bằng bộ lọc Kalman, các giá trị đo đưa vào véc-tơ \mathbf{z}_k có thể thu thập được từ một cảm biến như bộ lập mã trực quay hoặc được

kết hợp thêm từ nhiều cảm biến đo tuyệt đối khác. Các ma trận Q và A thuộc loại hệ thống phụ thuộc với một kết cấu robot, còn véc-tơ \mathbf{z} và các ma trận H , V và R sẽ có kích thước và giá trị khác nhau tùy vào số cảm biến ngoài được tổng hợp. Dưới đây sẽ xét lần lượt quá trình tính toán và kết quả thực nghiệm tổng hợp cảm biến với bộ lọc Kalman mở rộng trong 3 trường hợp với 3 cảm biến ngoài.

3.1.1. Tổng hợp dữ liệu odometry với cảm biến chỉ hướng từ-địa bàn.

3.1.1.1. Xác định các ma trận trong các bước tổng hợp cảm biến dùng EKF.

Vận tốc và quãng đường đi của robot trên hình 3.2 đo được bằng phương pháp odometry dựa trên hàm f trong phương trình hệ thống (3.1).

- *Khởi đầu* từ bước $k = 0$, robot xuất phát từ vị trí ban đầu được biết chính xác với ước lượng trạng thái hậu nghiệm $\hat{\mathbf{x}}_0$ và hiệp phương sai của ước lượng sai số P_0 .

- *Trong pha dự đoán:*

Trạng thái tiên nghiệm $\hat{\mathbf{x}}_k^-$ được tính theo phương trình (3.4) với các tín hiệu điều khiển đầu vào $\mathbf{u} = [\Delta s_R, \Delta s_L]^T$ được đo bằng cảm biến lập mã trục xoay.

Để nhận được ma trận P của bộ lọc, các ma trận Q và A được xác định như sau:

- Q_k là ma trận hiệp phương sai nhiễu lồi vào có kích thước $[2 \times 2]$, ta có:

$$Q_k = \text{covar}(\Delta s_{R(k)}, \Delta s_{L(k)}) = \begin{bmatrix} \delta_R | \Delta s_{R(k)} | & 0 \\ 0 & \delta_L | \Delta s_{L(k)} | \end{bmatrix} \quad (3.9)$$

Ở đây δ_R và δ_L là các hằng số sai số đại diện cho các tham số bất định của mạch điều khiển mô-tơ và tương tác của bánh xe với mặt sàn và chúng là độc lập với nhau. Như vậy, phương sai của nhiễu quá trình tỷ lệ thuận với trị tuyệt đối của các số gia quãng đường đi được bởi mỗi bánh xe Δs_R và Δs_L . Các hằng số tỷ lệ này phụ thuộc vào kết cấu của robot và môi trường. Chúng được xác định bằng thực nghiệm như sau: Đo sai số giữa vị trí thực và vị trí ước lượng của mô hình động học, khi điều khiển robot đi thẳng về phía trước nhiều lần (với vận tốc dài của robot tăng từ nhỏ nhất đến cực đại). Đo sai số giữa hướng thực và hướng ước lượng của mô hình động học của robot, khi điều khiển robot quay xung quanh một trục của nó

nhiều lần (với vận tốc góc quay của robot tăng từ nhỏ nhất đến cực đại). Dựa trên những sai số vị trí và hướng này, các tham số δ của ma trận hiệp phương sai lỗi vào Q_k được tính toán được xác định là $\delta_R = \delta_L \equiv \delta = 0,0003$.

- A_k là ma trận có kích thước $[3 \times 3]$, tính được bằng:

$$A_k = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial \theta} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial \theta} \\ \frac{\partial f_\theta}{\partial x} & \frac{\partial f_\theta}{\partial y} & \frac{\partial f_\theta}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s_k \cdot \sin(\theta_{k-1} + \Delta \theta_k / 2) \\ 0 & 1 & \Delta s_k \cdot \cos(\theta_{k-1} + \Delta \theta_k / 2) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

- W_k là ma trận có kích thước $[3 \times 2]$, tính được bằng:

$$W_k = \begin{bmatrix} \frac{\partial f_x}{\partial \varepsilon_R} & \frac{\partial f_x}{\partial \varepsilon_L} \\ \frac{\partial f_y}{\partial \varepsilon_R} & \frac{\partial f_y}{\partial \varepsilon_L} \\ \frac{\partial f_\theta}{\partial \varepsilon_R} & \frac{\partial f_\theta}{\partial \varepsilon_L} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_x}{\partial \Delta s_R} & \frac{\partial f_x}{\partial \Delta s_L} \\ \frac{\partial f_y}{\partial \Delta s_R} & \frac{\partial f_y}{\partial \Delta s_L} \\ \frac{\partial f_\theta}{\partial \Delta s_R} & \frac{\partial f_\theta}{\partial \Delta s_L} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \cos(\theta_{k-1} + \frac{\Delta \theta_k}{2}) + \frac{\Delta s_k}{2L} \sin(\theta_{k-1} + \frac{\Delta \theta_k}{2}) & \frac{1}{2} \cos(\theta_{k-1} + \frac{\Delta \theta_k}{2}) + \frac{\Delta s_k}{2L} \sin(\theta_{k-1} + \frac{\Delta \theta_k}{2}) \\ \frac{1}{2} \sin(\theta_{k-1} + \frac{\Delta \theta_k}{2}) + \frac{\Delta s_k}{2L} \cos(\theta_{k-1} + \frac{\Delta \theta_k}{2}) & \frac{1}{2} \sin(\theta_{k-1} + \frac{\Delta \theta_k}{2}) - \frac{\Delta s_k}{2L} \cos(\theta_{k-1} + \frac{\Delta \theta_k}{2}) \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \quad (3.11)$$

• *Trong pha hiệu chỉnh:*

Trong trường hợp này, ta dùng một phép đo tuyệt đối góc hướng của robot bằng cảm biến từ-địa bàn để tổng hợp qua bộ lọc Kalman với giá trị ước lượng vị trí bằng phương pháp odometry nói trên. Véc-tơ đo \mathbf{z}_k dùng một thành phần là số đo góc hướng φ_k nhận được từ cảm biến:

$$\mathbf{z}_k = [\varphi_k] \text{ trong đó } \varphi_k = \theta_k + \varepsilon_\varphi \quad (3.12)$$

với θ_k là góc hướng của robot và ε_φ là ồn đo của cảm biến từ-địa bàn.

Để xác định được hệ số lọc Kalman K_k , các ma trận R , H và V cần được tính. Ta gọi các ma trận H , V , R trong trường hợp này là H_{odm} , V_{odm} và R_{odm} .

Khi ẩn đi chỉ số k , lúc này ta có các giá trị ma trận H , V và R như sau:

- H_{odm} có kích thước $[1 \times 3]$, tính được bằng:

$$H_{odm} = \begin{bmatrix} \frac{\partial \varphi}{\partial x} & \frac{\partial \varphi}{\partial y} & \frac{\partial \varphi}{\partial \theta} \end{bmatrix} = [0 \quad 0 \quad 1] \quad (3.13)$$

- V_{odm} có kích thước $[1 \times 1]$, tính được bằng:

$$V_{odm} = \begin{bmatrix} \frac{\partial \varphi}{\partial \varepsilon_{\varphi}} \end{bmatrix} = [1] \quad (3.14)$$

- R_{odm} có kích thước $[1 \times 1]$ và bằng phương sai của ồn đo từ cảm biến:

$$R_{odm} = [var(\varphi)] \quad (3.15)$$

Để đơn giản, cảm biến được đặt ở vị trí trùng với phương hình chiếu của tâm robot trên mặt phẳng sàn nhà và hướng của khung cảm biến được giữ cố kết với hướng robot. Các số liệu về góc hướng ban đầu của cảm biến và góc hướng ban đầu của robot, so với phương từ trường quả đất, được ghi lại trong bộ nhớ robot và được dùng làm số liệu bù trừ trong các phép đo góc hướng tiếp theo của robot so với trục tọa độ phòng.

Nếu căn cứ theo số liệu kỹ thuật của nhà sản xuất, độ lệch chuẩn của phép đo góc của cảm biến từ-địa bàn là $0,1^\circ$ ta có thể ước lượng được ngay giá trị phương sai $var(\varphi)$ của cảm biến này là bằng $3,04 \times 10^{-6} \text{ rad}^2$. Tuy nhiên trong thực tế, do ảnh hưởng của từ trường quả đất và các vật nhiễm từ tại các địa điểm robot hoạt động là khác nhau nên phương sai của phép đo này đã được chúng tôi kiểm chuẩn lại. Bằng cách so sánh các mẫu đo của từ - địa bàn gắn trên robot với một phép đo tuyệt đối khác, robot được điều khiển quay quanh trục 360° với các vận tốc quay khác nhau, mỗi bước dịch chuyển là 10° trong môi trường hoạt động của robot.

Từ đó có kết quả: $var(\varphi)$ có giá trị thực tế gần bằng $2,79 \times 10^{-4} \text{ rad}^2$, tức là độ lệch chuẩn phải cỡ 1° .

3.1.1.2. Thực nghiệm và thảo luận.

a) *Thiết lập thực nghiệm.*

- Robot đã được thiết kế, chế tạo và được chạy thử kiểm tra các chỉ tiêu kỹ thuật trên mặt sàn phẳng.

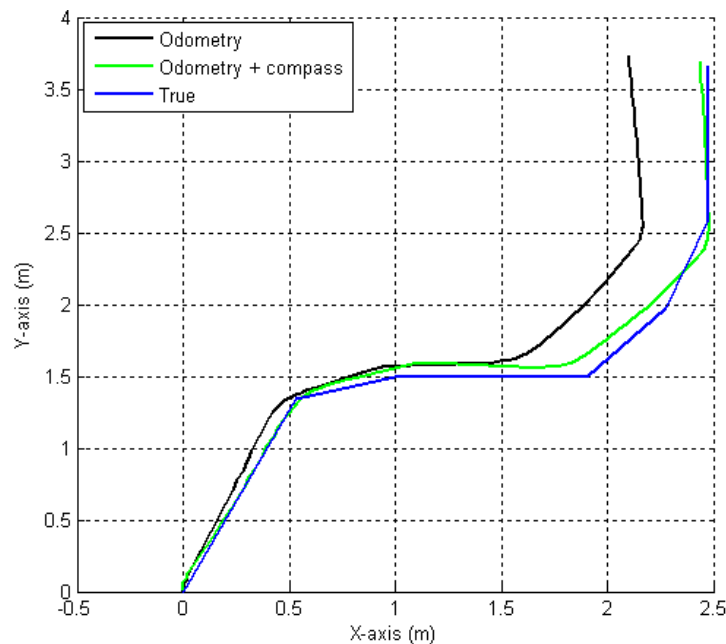
- Với thời gian lấy mẫu của bộ lọc là $\Delta t = 100 \text{ ms}$, phần mềm điều khiển và thu thập dữ liệu ước lượng trạng thái robot (khi áp dụng phương pháp tổng hợp cảm biến với EKF) được viết trong Visual C++. Dữ liệu được lưu trữ vào file số liệu trong bộ nhớ và hiển thị đồ họa lên màn hình máy tính.

- Hệ số tỷ lệ trong ma trận Q được xác định bằng thực nghiệm là $\delta = 0,0003$ và ma trận R có giá trị bằng phương sai $\text{var}(\varphi) = 2,79 \times 10^{-4} \text{rad}^2$ như nói trên.

- Đường đi thực của robot được đo bằng các thước đo thực tế tại mặt sàn.

b) Kết quả và thảo luận

Kết quả thu thập quỹ đạo mà robot đi được trong 3 trường hợp được biểu diễn trên hình 3.3 với đường đi thực (màu xanh dương), đường đi được ước lượng chỉ bằng phương pháp odometry (màu đen) và đường đi được ước lượng khi có dùng bộ lọc EKF với cảm biến từ-địa bàn (màu xanh lá cây). Bằng trực quan thấy ngay đường đi được ước lượng với bộ lọc EKF gần với đường đi thực hơn so với trường hợp chỉ dùng phương pháp odometry. Tính toán định lượng sẽ được báo cáo sau.



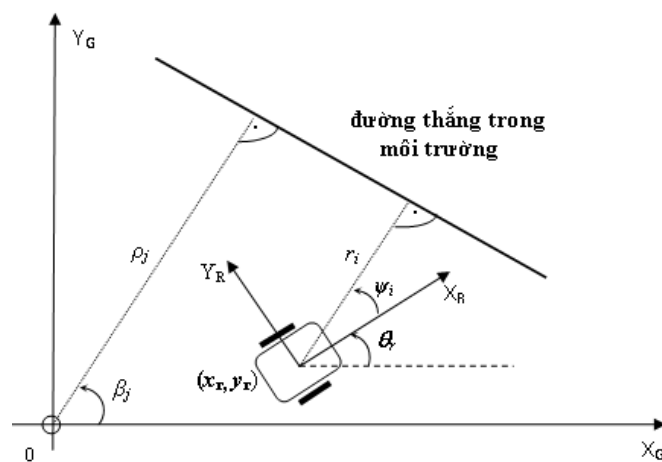
Hình 3.3. Quỹ đạo ước lượng của robot với các cấu hình không có (màu đen) và có EKF (màu xanh lá cây) so với quỹ đạo thực (màu xanh dương).

3.1.2. Tổng hợp dữ liệu với thêm cảm biến đo xa laser.

Cũng trên hệ thống robot trên, chúng tôi sử dụng thêm một phép đo tuyệt đối nữa từ cảm biến đo xa laser LRF được tổng hợp với bộ lọc EKF nhằm nâng cao hơn

nửa độ chính xác của phép ước lượng vị trí. Do vẫn được thực nghiệm trên cùng một hệ thống robot nên các giá trị ma trận A , Q và W được giữ nguyên như trên. Còn các ma trận H , V và R phải được cập nhật.

Máy đo xa laser LRF cho phép phát hiện các điểm đặc trưng của môi trường quanh robot. Trong trường hợp này chúng tôi chọn đó là các đoạn thẳng được LRF phát hiện trong môi trường (ví dụ như đường cắt ngang một bức tường phẳng trong phòng).



Hình 3.4 Máy đo LRF đặt trên robot đo 2 thông số của một đường thẳng trong môi trường.

Bằng phép biến đổi như biến đổi Hough [44], hai thông số của đường thẳng này có thể nhận được, đó là: khoảng cách r từ đường thẳng đến gốc tọa độ và góc ψ giữa đường thẳng và một trục tọa độ như hình 3.4 chỉ ra.

3.1.2.1. Tính véc-tơ z_k , ma trận H_k và ma trận hiệp phương sai R_k

Tại lần quét đầu tiên của LRF, một bản đồ toàn cục về môi trường được xây dựng bao gồm một tập hợp các đoạn thẳng được mô tả bởi các tham số β_j và ρ_j .

Phương trình đường thẳng dạng chính tắc là:

$$x_G \cos \beta_j + y_G \sin \beta_j = \rho_j \quad (3.16)$$

Trong đó, X_G và Y_G là các trục tọa độ toàn cục. Khi robot di chuyển, một dòng quét tia đo xa laser mới được thực hiện và một bản đồ cục bộ về môi trường được xây dựng với tọa độ gắn trên robot. Nó cũng bao gồm một tập hợp các đoạn thẳng được mô tả bởi phương trình:

$$x_R \cos \psi_i + y_R \sin \psi_i = r_i \quad (3.17)$$

Trong đó r_i và ψ_i là các tham số của đường thẳng thứ i trong số N đường như ở hình 3.4, với X_R và Y_R là các trục tọa độ cục bộ gắn với robot. Các đoạn thẳng của môi trường cục bộ và toàn cục sẽ được phát hiện làm khớp với nhau khi sử dụng giải thuật “weighted line fitting” [101], như sẽ được trình bày ở phần sau.

Làm khớp các tham số đường thẳng ở bản đồ cục bộ sẽ được thu thập vào trong véc-tơ \mathbf{z}_k , được sử dụng như đầu vào cho bước chỉnh sửa của EKF để cập nhật trạng thái của robot.

$$\mathbf{z}_k = [\varphi_k, r_{1,k}, \psi_{1,k}, \dots, r_{N,k}, \psi_{N,k}]^T \quad (3.18)$$

Các ma trận H , V và R cần được bổ sung giá trị với tên đặt lại cho cấu hình mới là H_{odml} , V_{odml} và R_{odml}

Từ tọa độ và hướng của robot được ước tính bởi phương pháp odometry, các tham số ρ_j và β_j của đoạn thẳng thứ j trong bản đồ toàn cục được chuyển thành các tham số r_i và ψ_i theo hệ tọa độ cục bộ của robot và được tính bằng:

$$\begin{bmatrix} r_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} |C_j| \\ \beta_j - \theta + (-0,5 \operatorname{sign}(C_j) + 0,5)\pi \end{bmatrix} \quad (3.19)$$

trong đó, $C_j = \rho_j - x \cdot \cos \beta_j - y \cdot \sin \beta_j$

Các phép đo r_i và ψ_i là các phép đo trực tiếp chịu các nhiễu đo của cảm biến laser LRF. Gọi các nhiễu đo này là ε_r và ε_ψ thì véc-tơ thành phần đo từ cảm biến laser khi tính cả các can nhiễu sẽ là:

$$\begin{bmatrix} r_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} |C_j| \\ \beta_j - \theta_r + (-0,5 \operatorname{sign}(C_j) + 0,5)\pi \end{bmatrix} + \begin{bmatrix} \varepsilon_r \\ \varepsilon_\psi \end{bmatrix} \quad (3.20)$$

Như vậy, so với khi chỉ dùng 1 cảm biến ngoài là từ-địa bàn thì khi có thêm cảm biến LRF hàm đo h được bổ sung thêm 2N thành phần r_i và ψ_i . Khi ẩn chỉ số k , ta có các ma trận như sau:

- Ma trận H_{odml} có kích thước $[(1+2N) \times 3]$ và bằng:

$$H_{odml} = \begin{bmatrix} \frac{\partial \varphi}{\partial x} & \frac{\partial \varphi}{\partial y} & \frac{\partial \varphi}{\partial \theta} \\ \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \theta} \\ \frac{\partial \psi_1}{\partial x} & \frac{\partial \psi_1}{\partial y} & \frac{\partial \psi_1}{\partial \theta} \\ \dots & \dots & \dots \\ \frac{\partial r_N}{\partial x} & \frac{\partial r_N}{\partial y} & \frac{\partial r_N}{\partial \theta} \\ \frac{\partial \psi_N}{\partial x} & \frac{\partial \psi_N}{\partial y} & \frac{\partial \psi_N}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -c_1 \cos(\beta_1) & -c_1 \sin(\beta_1) & 0 \\ 0 & 0 & -1 \\ \dots & \dots & \dots \\ -c_1 \cos(\beta_N) & -c_1 \sin(\beta_N) & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (3.21)$$

Ở đây c_i là các hàm dấu $sign(C_i)$.

- Ma trận V_{odml} có kích thước $[(1+2N) \times 3]$ và bằng:

$$V_{odml} = \begin{bmatrix} \frac{\partial \varphi}{\partial \varepsilon_\varphi} & \frac{\partial \varphi}{\partial \varepsilon_r} & \frac{\partial \varphi}{\partial \varepsilon_\psi} \\ \frac{\partial r_1}{\partial \varepsilon_\varphi} & \frac{\partial r_1}{\partial \varepsilon_r} & \frac{\partial r_1}{\partial \varepsilon_\psi} \\ \frac{\partial \psi_1}{\partial \varepsilon_\varphi} & \frac{\partial \psi_1}{\partial \varepsilon_r} & \frac{\partial \psi_1}{\partial \varepsilon_\psi} \\ \dots & \dots & \dots \\ \frac{\partial r_N}{\partial \varepsilon_\varphi} & \frac{\partial r_N}{\partial \varepsilon_r} & \frac{\partial r_N}{\partial \varepsilon_\psi} \\ \frac{\partial \psi_N}{\partial \varepsilon_\varphi} & \frac{\partial \psi_N}{\partial \varepsilon_r} & \frac{\partial \psi_N}{\partial \varepsilon_\psi} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \dots & \dots & \dots \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

- Ma trận nhiễu đo R_{odml} có kích thước $[(1+2N) \times (1+2N)]$:

$$R_{odml} = \begin{bmatrix} var(\varphi) & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & var(r_1) & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & var(\psi_1) & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & var(r_N) & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & var(\psi_N) \end{bmatrix} \quad (3.23)$$

Các giá trị phương sai của các phép đo từ địa bàn đã được tính như nêu ở trên, các giá trị còn lại thuộc các phép đo của cảm biến đo xa laser ($r_i \psi_i$) sẽ được trình bày tính ở phần dưới. Các số liệu này sẽ được điền vào cho bước điều chỉnh EKF.

3.1.2.2. Nhận dạng các tham số $var(r_i)$ và $var(\psi_i)$ của ma trận R_k .

Để có được các tham số của ma trận R_k như nói trên, các bước sau cần được tiến hành:

a) *Phát hiện các đoạn thẳng.*

Nội dung phần này mô tả thuật toán tách các đoạn thẳng từ tập hợp các điểm ảnh quét dựa vào phương pháp bình phương tối thiểu. Các điểm quét được thu về từ LRF với góc quét 180° và độ phân giải 0.5° .

Trước tiên, dữ liệu thu được từ LRF được chuyển từ tọa độ cực về tọa độ Đề-cac khi sử dụng công thức (3.24).

$$x_i = r_i \cos\left(\beta_i \frac{\pi}{180}\right); \quad y_i = r_i \sin\left(\beta_i \frac{\pi}{180}\right) \quad (3.24)$$

Mỗi tập (r_i, β_i) là số liệu các điểm ảnh nhận được từ LRF. Do LRF quét từ trái qua phải, các điểm quét gần nhau sẽ có nhiều khả năng nằm trên cùng một đối tượng ví dụ như: bức tường, bảng,... Phương pháp bình phương tối thiểu được áp dụng lên một nhóm gồm N_1 điểm quét liên kế nhau tính từ điểm quét đầu tiên để tìm ra đường thẳng tốt nhất đi qua những điểm quét này. Một ngưỡng khoảng cách $disT$ sẽ được xác định trước và so sánh với khoảng cách từ mỗi điểm trong nhóm tới đường thẳng tốt nhất này. Nếu như có nhiều hơn N_2 điểm thỏa mãn điều kiện khoảng cách nhỏ hơn $disT$, ta có thể coi như xác định được một đường thẳng và một đường thẳng tốt nhất mới sẽ được tính dựa trên N_2 điểm này. Điểm có chỉ số thấp nhất sẽ được chọn làm đầu mút của đoạn thẳng. Hệ số góc (slope) và điểm cắt (intercept) của đường thẳng chứa đoạn thẳng cần tách được tính theo công thức (3.25).

$$m = \frac{\sum_{i=1}^n x_i y_i - \frac{\left(\sum_{i=1}^n x_i\right)\left(\sum_{i=1}^n y_i\right)}{n}}{\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i\right)^2}{n}}; \quad k = \bar{y} - m\bar{x} \quad (3.25)$$

Trong đó n là tổng số điểm quét trên đường thẳng, \bar{x} và \bar{y} là các giá trị trung bình của x và y tương ứng. Thuật toán được áp dụng lên những điểm quét còn lại để xác định những điểm tiếp theo thuộc đoạn thẳng vừa được tách, các điểm cách đoạn thẳng quá xa sẽ bị loại bỏ. Điểm cuối cùng thỏa mãn điều kiện nhỏ hơn ngưỡng khoảng cách sẽ được xét làm đầu mút còn lại của đoạn thẳng. Áp dụng thuật toán với các nhóm điểm còn lại cho đến khi tất cả các điểm quét đều được kiểm tra.

Tuy nhiên sử dụng một ngưỡng khoảng cách cố định không đảm bảo tính chính xác cho tất cả các lần quét. Kết quả thực nghiệm cho thấy có khoảng 10% số lần quét yêu cầu ngưỡng khoảng cách lớn hơn để có thể tách ra được các đường thẳng. Mặt khác nếu ngưỡng khoảng cách quá lớn sẽ dẫn đến kết quả các đoạn thẳng tách ra không được mịn và gây ảnh hưởng đến kết quả khớp đoạn. Nghiên cứu này phát triển một ngưỡng khoảng cách động đơn giản như sau: trước tiên xác định một ngưỡng lớn nhất $disTmax$ để xác định xem một điểm quét có thuộc một đoạn thẳng hay không. Tuy nhiên đối với mỗi nhóm điểm, khoảng cách từ các điểm quét đến đường thẳng tốt nhất sẽ được so sánh với nhau và sắp xếp tăng dần thay vì so sánh với $disT$. Nếu điểm thứ N_2 trong nhóm sau khi sắp xếp có khoảng cách nhỏ hơn $disTmax$, khoảng cách đó sẽ được chọn làm ngưỡng cho đoạn thẳng được tách. Như vậy, ứng với số đoạn thẳng tách được ta sẽ có số ngưỡng khoảng cách tương đương.

b) *Khớp các đoạn cục bộ và toàn cục.*

Các đoạn thẳng thể hiện cùng một đối tượng trên bản đồ cục bộ và bản đồ toàn cục sẽ được khớp với nhau khi sử dụng thuật toán được mô tả dưới đây. Gọi G_j và L_i là các đoạn thẳng tương ứng trong bản đồ toàn cục và bản đồ cục bộ.

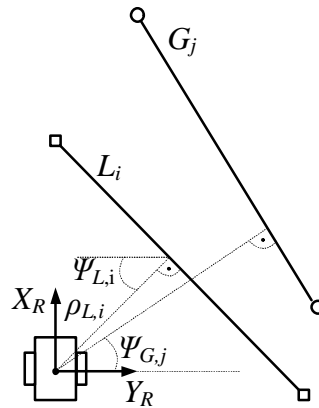
$$\begin{aligned} G_j &= [x_{1G,j} \quad x_{2G,j} \quad y_{1G,j} \quad y_{2G,j}] & j &= 1 \dots n_G \\ L_i &= [x_{1L,i} \quad x_{2L,i} \quad y_{1L,i} \quad y_{2L,i}] & i &= 1 \dots n_L \end{aligned} \quad (3.26)$$

Trong đó $(x_1, y_1), (x_2, y_2)$ là tọa độ hai điểm mút của mỗi đoạn thẳng tương ứng, n_G và n_L là số đoạn thẳng tách được trong bản đồ toàn cục và bản đồ cục bộ. Các đoạn thẳng thuộc bản đồ toàn cục sẽ được chuyển về bản đồ cục bộ theo công thức sau:

$$G_{Tj} = G \left(\begin{bmatrix} x_{1G,j} & y_{1G,j} \\ x_{2G,j} & y_{2G,j} \end{bmatrix} - \begin{bmatrix} x_0 & y_0 \\ x_0 & y_0 \end{bmatrix} \right) \quad (3.27)$$

Trong đó G là ma trận chuyển, $[x_o \ y_o \ \phi_o]^T$ biểu diễn vị trí toàn cục của robot xác định theo phương pháp odometry. Mỗi đoạn thẳng cục bộ L_i sẽ được so sánh với tất cả các đoạn thẳng toàn cục đã chuyển về bản đồ cục bộ. Hai đoạn thẳng được coi là “khớp” với nhau nếu khoảng cách và góc định hướng từ robot tới chúng là tương đương và mức trùng khớp giữa hai đoạn thẳng nhỏ hơn một ngưỡng cho trước.

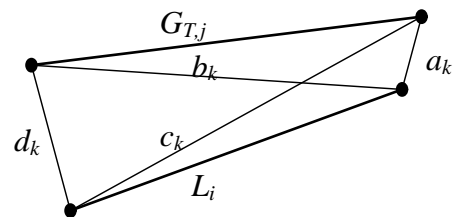
Khoảng cách và góc định hướng từ các đoạn thẳng toàn cục và cục bộ tới vị trí robot được biểu diễn theo $(\rho_{G,j}, \psi_{G,j})$ và $(\rho_{L,i}, \psi_{L,i})$ tương ứng.



Hình 3.5 Các tham số (ρ, ψ) của các phân đoạn đường thẳng với vị trí robot trong hệ tọa độ của robot.

Mức trùng khớp giữa đoạn thẳng cục bộ L_i và đoạn thẳng toàn cục được chuyển về bản đồ cục bộ $G_{T,i}$ được định nghĩa như sau:

$$\begin{aligned} O_k(a_k, b_k) &= |a_k + b_k - \overline{G_{T,j}}| \\ O_k(c_k, d_k) &= |c_k + d_k - \overline{G_{T,j}}| \\ k &= (j-1)n_L + i \\ k &= 1, 2, \dots, n_L n_G - 1, n_L n_G \end{aligned}$$



Hình 3.6 Các tham số trùng khớp giữa các phân đoạn đường thẳng cục bộ và toàn cục.

Trong đó $\overline{G_{T,j}}$ là độ dài đoạn thẳng toàn cục đã được chuyển về bản đồ cục bộ, a_k, b_k, c_k và d_k là khoảng cách giữa các điểm mút của hai đoạn thẳng L_i và $G_{T,j}$

Các bất đẳng thức dưới đây trình bày điều kiện khớp hai đoạn thẳng toàn cục và cục bộ trong hệ tọa độ của robot, hai đoạn thẳng L_i và $G_{T,j}$ được coi là khớp nhau nếu tất cả các điều kiện sau được thỏa mãn.

$$\begin{cases} O_k(a_k, b_k) < T \\ O_k(c_k, d_k) < T \\ (\rho_{L,i} - \rho_{G,j})^2 < T_\rho \\ (\psi_{L,i} - \psi_{G,j})^2 < T_\psi \end{cases} \quad (3.28)$$

trong đó T, T_ρ, T_ψ là các ngưỡng định trước, xác định bằng thực nghiệm.

c) *Tính toán tham số cho ma trận R_k .*

Để tính sai số đánh giá cho EKF, mỗi đoạn thẳng được tách ra sẽ được biểu diễn sử dụng các thông số (ρ, ψ) trong đó ρ là khoảng cách từ robot đến đoạn thẳng và ψ là góc hướng đoạn thẳng.

$$\rho = \frac{|-k|}{\sqrt{m^2 + 1}}; \psi = \arctan 2 \left(\frac{k}{1 + m^2}, \frac{-mk}{1 + m^2} \right) \quad (3.29)$$

Trong đó (m, k) là hệ số góc và điểm cắt của các đường thẳng tốt nhất được tách theo các công thức (3.25). Khoảng cách từ robot đến đoạn thẳng có thể được tính trong hệ tọa độ cục bộ như sau:

$$d_i = r_i \cos(\psi_i - \psi) - \rho \quad (3.30)$$

Đường thẳng tốt nhất được tính dựa trên việc xác định tổng bình phương của các khoảng cách từ điểm quét đến đường thẳng là nhỏ nhất:

$$\operatorname{argmin}_{(\rho, \psi)} \sum_i (r_i \cos(\psi_i - \psi) - \rho)^2 \quad (3.31)$$

Theo phương pháp trình bày bởi [35] và Deriche [45], ma trận R có thể được tính dựa trên giả thiết mỗi điểm dữ liệu đều có sai số trong hệ tọa độ Đề-cac là như nhau:

$$R_i = \frac{a_i \sigma_{yy}^2 - b_i \sigma_{xy}^2 + c_i \sigma_{xx}^2}{(a_i - c_i)^2 + b_i^2} \begin{pmatrix} 1 & -e \\ -e & e^2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\sigma_{yy}^2 \cos^2 \varphi_i + \sigma_{xx}^2 \sin^2 \varphi_i - 2\sigma_{xy}^2 \sin \varphi_i \cos \varphi_i}{n} \end{pmatrix} \quad (3.32)$$

$$\cong \begin{vmatrix} \text{var}(r_i) & 0 \\ 0 & \text{var}(\psi_i) \end{vmatrix}$$

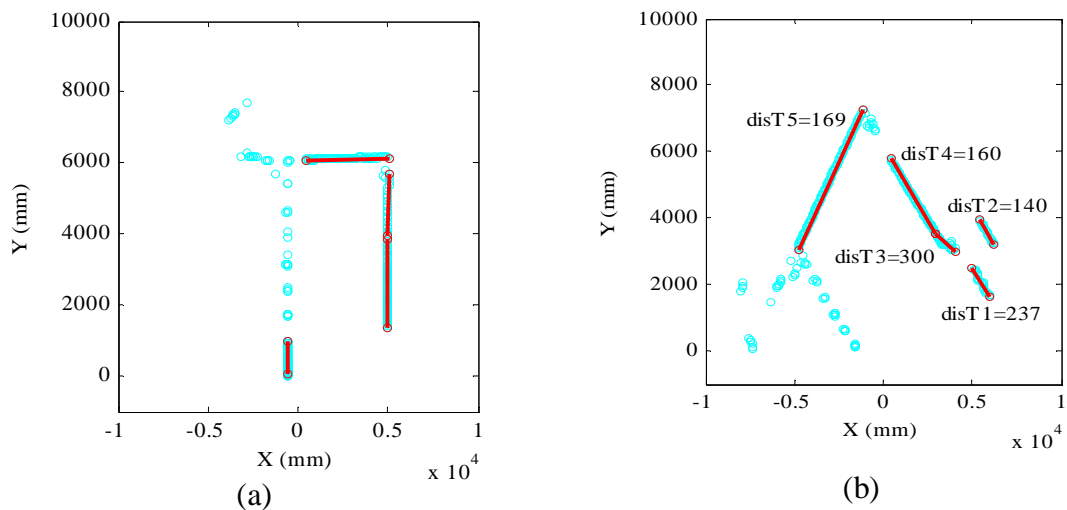
Trong đó:

$$e_i = \bar{y} \cos \psi_i - \bar{x} \sin \psi_i; \quad \varphi_i = \left(\psi_i + \frac{\pi}{x} \right); \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i; \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i;$$

$$a_i = \sum_{i=1}^n (x_i - \bar{x})^2; b_i = 2 \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}); c_i = \sum_{i=1}^n (y_i - \bar{y})^2.$$

3.1.2.3. Thực nghiệm và thảo luận.

a) *Công đoạn trích chọn các đoạn thẳng*: Thuật toán đã được lập trình trong MATLAB và thời gian tính toán cho một tập dữ liệu 360 điểm với ngưỡng cố định là 150ms; ngưỡng động là 200ms ÷ 250ms tùy thuộc vào số lượng các đoạn thẳng được phát hiện.



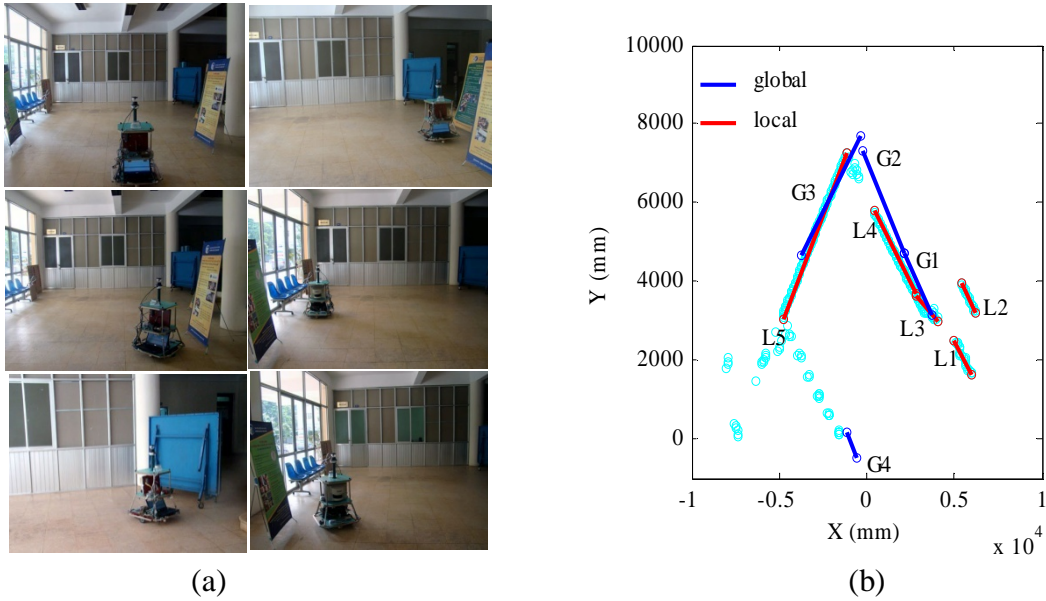
Hình 3.7 a) Trích chọn các đoạn thẳng với ngưỡng cố định; b) Trích chọn các đoạn thẳng với ngưỡng động.

Hình 3.7a cho thấy kết quả với ngưỡng cố định, hình 3.7b cho thấy kết quả với các ngưỡng khác nhau cho các đoạn thẳng khác nhau.

b) *Công đoạn làm khớp các bản đồ toàn cục và cục bộ*.

Kết quả làm khớp các đoạn thẳng từ bản đồ toàn cục (các đường thẳng màu xanh da trời) vào bản đồ cục bộ (màu đỏ) như biểu diễn trên hình 3.8b.

Trong thực tế, có những đoạn thẳng được phát hiện trong bản đồ toàn cục, nhưng không xuất hiện trong các bản đồ cục bộ và ngược lại. Vì vậy, trong trường hợp này, tác giả chỉ nghiên cứu những đoạn thẳng xuất hiện trong cả hai bản đồ. Với các tiêu chí phù hợp được xem xét, (L3, G1), (L4, G2), (L5, G3) đáp ứng tất cả các điều kiện và được khớp cặp với nhau.

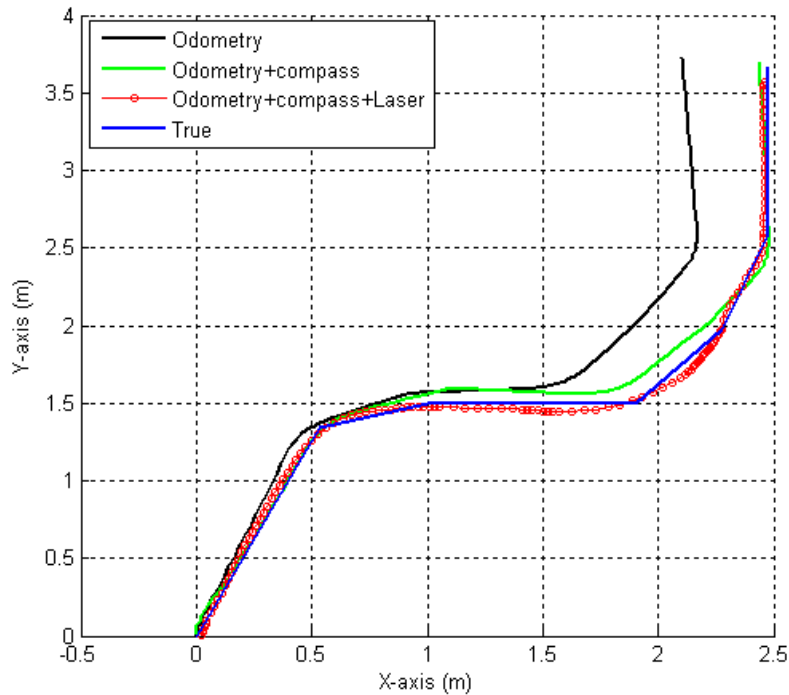


Hình 3.8 a) Ảnh chụp liên tiếp robot chuyển động trong môi trường; b) Làm khớp các đoạn thẳng ở bản đồ toàn cục và bản đồ cục bộ.

Tương tự như ở trên, thời gian lấy mẫu Δt của EKF là 100ms; các giá trị $var(r_i)$, $var(\psi_i)$ của R_k sẽ được tính như sau: Từ tập số liệu thu thập được từ môi trường của LRF (hình 3.8b), chúng ta thấy thuật toán phát hiện ra 6 đoạn thẳng. Áp dụng công thức (3.32) và sử dụng phần mềm Matlab để tính toán, kết quả các giá trị $var(r_i)$, $var(\psi_i)$ của R_k trên bảng 3.1.

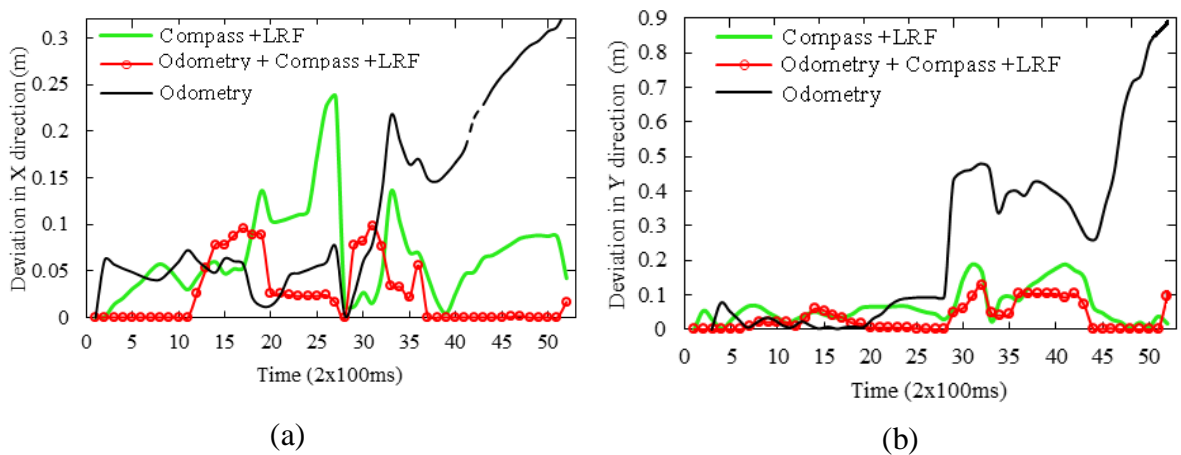
Bảng 3.1. Giá trị của R_k của LRF.

I	var(r_i) (m²)	var(ψ_i) (rad²)
1	0,0001	0,15943
2	0,0006	0,07248
3	0,0001	0,08895
4	0,0001	0,01077
5	0,0001	0,00168
6	0,0002	0,00010



Hình 3.9 Quỹ đạo ước tính của robot với các cấu hình EKF khác nhau.

Kết quả khi so sánh với đường đi thực của robot (xanh dương) cho thấy việc áp dụng tổng hợp thêm dữ liệu của các cảm biến đo tuyệt đối cho một sự cải thiện đáng kể độ chính xác của các giá trị ước lượng. Hình 3.10 dưới đây biểu diễn các giá trị sai lệch theo phương X và phương Y giữa các vị trí được ước tính với đường thực.



Hình 3.10 Độ lệch giữa các vị trí được ước tính với đường thực.
a.) Theo phương X; b) Theo phương Y.

3.1.3. Tổng hợp dữ liệu với cảm biến camera ảnh toàn phương.

3.1.3.1. Tính véc tơ z_k , ma trận H_k và ma trận hiệp phương sai R_k .

Cũng giống như trường hợp ở trên, nhưng trong trường hợp này thêm vào một cảm biến ảnh toàn phương (omni-directional camera). Cảm biến này cho phép thêm vào một phép đo góc γ_k giữa hướng của robot và một cột mốc có tọa độ biết trước để có thể tổng hợp với các phép ước lượng như đã nêu ở trên. Véc-tơ z_k lúc này sẽ có giá trị:

$$\mathbf{z}_k = [\varphi_k, r_1, \psi_1, \dots, r_N, \psi_N, \gamma_k]^T \quad (3.33)$$

Ảnh toàn phương chứa đựng thông tin về vị trí của các đối tượng theo các góc hướng trong toàn không gian 360° [55]. Nếu ta sử dụng camera toàn phương cho một phép đo định hướng tuyệt đối và tổng hợp giá trị này với các phép đo khác thì cũng tạo ra được một hệ thống ước lượng vị trí tốt hơn cho robot. Chúng tôi chọn phép đo tuyệt đối đó là đo góc hướng γ_k giữa robot và một điểm (cột) mốc được phát hiện bởi camera.

a) *Nhận dạng phép đo góc từ cảm biến ảnh toàn phương.*

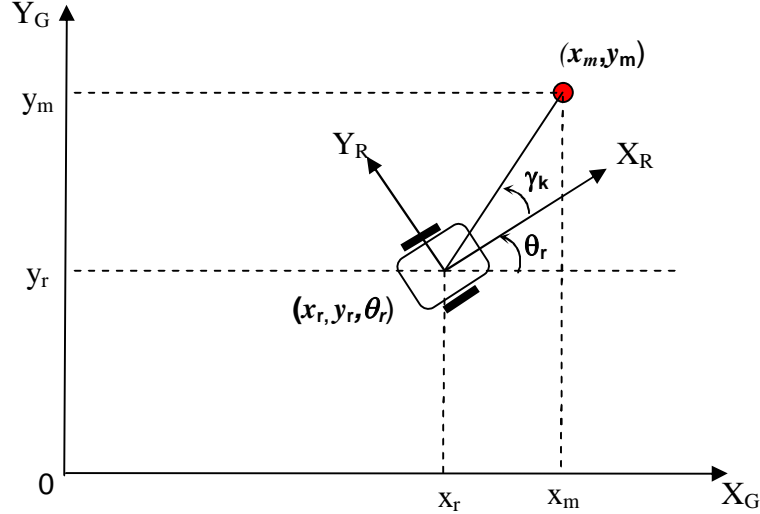
Phương pháp này dựa trên việc phát hiện một cột mốc màu đỏ nằm ở một vị trí tọa độ biết trước (x_m, y_m) và được thực hiện như mô tả ở hình 3.11 (kỹ thuật phát hiện cột mốc sẽ được trình bày ở mục sau).

Tại bước chuyển động k , vị trí của robot được ước lượng bởi phép đo odometry là (x_k, y_k, θ_k) trong đó lưu ý θ_k là góc lệch của trục tọa độ cục bộ X_R của robot so với trục tọa độ toàn cục X_G . Lưu ý, theo quy ước thông thường góc nhận giá trị dương khi có chiều theo ngược chiều kim đồng hồ. Góc hướng γ_k của robot so với cột mốc là góc giữa đường thẳng nối tâm robot (cũng là tâm camera) với cột mốc và trục tọa độ X_R . Theo hình 3.11, góc γ_k mà cảm biến camera đo được cũng được xác định qua phương pháp odometry và tọa độ cột mốc biết trước theo công thức sau:

$$\tan(\gamma_k + \theta_k) = \frac{y_m - y_k}{x_m - x_k}$$

Gọi nhiễu đo camera là ε_γ , khi tính cả nhiễu ta có góc hướng nhận được là:

$$\gamma_k = \arctan\left(\frac{y_m - y_k}{x_m - x_k}\right) - \theta_k + \varepsilon_\gamma \quad (3.34)$$



Hình 3.11. Ước lượng góc nhìn γ_k từ robot đến vật mốc màu đỏ (x_m, y_m) bằng phép đo odometry và cảm biến ảnh toàn phương.

Giá trị γ_k này được bổ sung vào véc-tơ \mathbf{z}_k của bộ lọc Kalman, cho phép nâng cao độ tin cậy của phép định vị.

Các ma trận H , V và R trong cấu hình này được gọi là H_{odmlc} , V_{odmlc} và R_{odmlc} .

Thực hiện các phép lấy đạo hàm riêng của γ_k theo các biến trạng thái x , y , θ và theo các nhiễu ε_φ , ε_r , ε_ψ và ε_γ . Khi ẩn đi chỉ số k ta có các ma trận được xác định như sau:

- H_{odmlc} có kích thước $[(1+2N+1) \times 3]$ và bằng:

$$H_{odmlc} = \begin{bmatrix} \frac{\partial \varphi}{\partial x} & \frac{\partial \varphi}{\partial y} & \frac{\partial \varphi}{\partial \theta} \\ \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \theta} \\ \frac{\partial \psi_1}{\partial x} & \frac{\partial \psi_1}{\partial y} & \frac{\partial \psi_1}{\partial \theta} \\ \dots & \dots & \dots \\ \frac{\partial r_N}{\partial x} & \frac{\partial r_N}{\partial y} & \frac{\partial r_N}{\partial \theta} \\ \frac{\partial \psi_N}{\partial x} & \frac{\partial \psi_N}{\partial y} & \frac{\partial \psi_N}{\partial \theta} \\ \frac{\partial \gamma}{\partial x} & \frac{\partial \gamma}{\partial y} & \frac{\partial \gamma}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ -c_1 \cos(\beta_1) & -c_1 \sin(\beta_1) & 0 \\ 0 & 0 & -1 \\ \dots & \dots & \dots \\ -c_1 \cos(\beta_N) & -c_1 \sin(\beta_N) & 0 \\ 0 & 0 & -1 \\ \frac{(y_m - y)}{(y_m - y)^2 + (x_m - x)^2} & \frac{-(x_m - x)}{(y_m - y)^2 + (x_m - x)^2} & -1 \end{bmatrix} \quad (3.35)$$

trong đó c_i là các hàm dấu $sign(C_i)$.

- V_{odmlc} có kích thước $[(1+2N+1) \times 4]$ và bằng:

$$V_{odmlc} = \begin{bmatrix} \frac{\partial \varphi}{\partial \varepsilon_\varphi} & \frac{\partial \varphi}{\partial \varepsilon_r} & \frac{\partial \varphi}{\partial \varepsilon_\psi} & \frac{\partial \varphi}{\partial \varepsilon_\gamma} \\ \frac{\partial r_1}{\partial \varepsilon_\varphi} & \frac{\partial r_1}{\partial \varepsilon_r} & \frac{\partial r_1}{\partial \varepsilon_\psi} & \frac{\partial r_1}{\partial \varepsilon_\gamma} \\ \frac{\partial \psi_1}{\partial \varepsilon_\varphi} & \frac{\partial \psi_1}{\partial \varepsilon_r} & \frac{\partial \psi_1}{\partial \varepsilon_\psi} & \frac{\partial \psi_1}{\partial \varepsilon_\gamma} \\ \dots & \dots & \dots & \dots \\ \frac{\partial r_N}{\partial \varepsilon_\varphi} & \frac{\partial r_N}{\partial \varepsilon_r} & \frac{\partial r_N}{\partial \varepsilon_\psi} & \frac{\partial r_N}{\partial \varepsilon_\gamma} \\ \frac{\partial \psi_N}{\partial \varepsilon_\varphi} & \frac{\partial \psi_N}{\partial \varepsilon_r} & \frac{\partial \psi_N}{\partial \varepsilon_\psi} & \frac{\partial \psi_N}{\partial \varepsilon_\gamma} \\ \frac{\partial \gamma_N}{\partial \varepsilon_\varphi} & \frac{\partial \gamma_N}{\partial \varepsilon_r} & \frac{\partial \gamma_N}{\partial \varepsilon_\psi} & \frac{\partial \gamma_N}{\partial \varepsilon_\gamma} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.36)$$

- Ma trận R_k có kích thước $[(1+2N+1) \times (1+2N+1)]$ và bằng:

$$R_{odmlc} = \begin{bmatrix} var(\varphi) & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & var(r_1) & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & var(\psi_1) & 0 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & var(r_N) & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & var(\psi_N) & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & var(\gamma) \end{bmatrix} \quad (3.37)$$

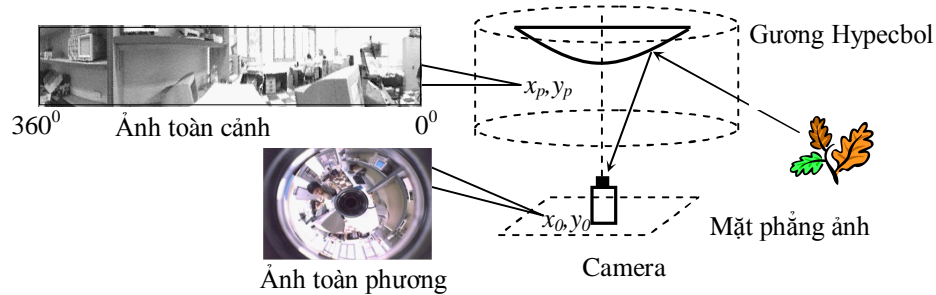
3.1.3.2. Áp dụng kỹ thuật thị giác máy tính phát hiện cột mốc và đo góc γ_k .

Đặc tính của cảm biến ảnh toàn phương là bảo toàn về góc hướng, hướng của ảnh so với phương chính diện bằng góc tạo bởi đường nối giữa tâm ảnh (tâm quang học) và trục thẳng đứng của hình. Do vậy thông thường người ta thực hiện chuyển đổi từ ảnh toàn phương (omnidirectional) thu nhận được sang ảnh toàn cảnh (panoramic) qua phép chiếu lên một mặt trụ tròn có tâm trùng với trục quang học giữa kính phản xạ và thấu kính. Sau đó tùy theo mục đích sử dụng ta sẽ thực hiện chiếu ảnh dùng phép chiếu phối cảnh lên mặt phẳng vuông góc theo hướng quan sát cần quan.

Từ tọa độ tâm ảnh toàn phương (x_c, y_c) như minh họa trên hình 3.12 và tọa

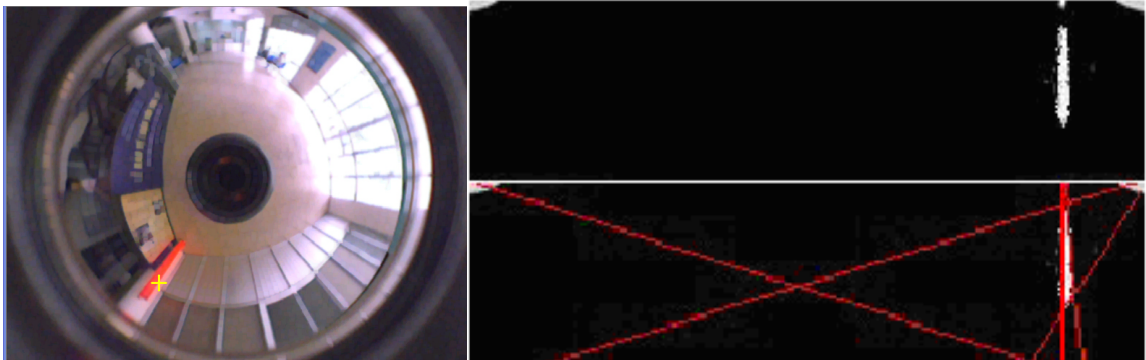
độ điểm ảnh toàn phương bất kỳ (x_0, y_0) có thể tìm được tọa độ tương ứng của ảnh toàn cảnh (x_p, y_p) là hình chiếu các đối tượng trong không gian trên một mặt trụ tròn có trục trùng với quang trục hệ thống camera như sau [55]:

$$\begin{aligned} x_0 &= x_c + r \sin \gamma \\ y_0 &= x_c + r \cos \gamma \end{aligned} \quad \text{Trong đó } r = y_p \text{ và } \gamma = \frac{x_p \cdot 360}{720} = \frac{x_p}{2} \quad (3.38)$$



Hình 3.12 Ảnh toàn phương và trải ảnh toàn cảnh của camera-omni.

Như vậy do đặc tính thú vị về bảo toàn hướng, tất cả các đường thẳng đứng (có ảnh đi qua tâm) ở trong ảnh toàn phương sau khi trải ảnh toàn cảnh thì nó cũng sẽ là đường thẳng đứng. Trong thực nghiệm, chúng tôi đã dựng một cột mốc màu đỏ được đặt theo phương thẳng đứng ở một tọa độ đã biết trước (x_m, y_m) , hình 3.13.



Hình 3.13. Phát hiện đường thẳng đứng sử dụng thuật toán Hough.

Chương trình phần mềm xử lý ảnh có nhiệm vụ xác định được góc tạo bởi một điểm đánh dấu trên robot (chính là của một điểm gốc được chọn trên ảnh camera toàn phương) và điểm chiếu của cột mốc lên mặt phẳng sàn nhà OXY. Nói ngắn gọn là xác định được góc giữa robot và cột mốc. Quá trình xử lý dùng công

nghệ thị giác máy tính với thư viện xử lý ảnh mã nguồn mở OpenCV đã được tiến hành qua các bước dưới đây.

Ảnh toàn phương được chuyển ra ảnh toàn cảnh theo công thức (3.38). Bằng các hàm trong OpenCV [28] phát hiện mảng ảnh cột mốc màu đỏ. Chuyển ảnh màu RGB ra ảnh đa mức xám, lọc nhiễu và chuyển sang ảnh nhị phân. Dùng giải thuật Hough để phát hiện tọa độ đoạn thẳng biểu diễn cột mốc. Căn cứ vào đó và vào điểm gốc trên ảnh camera, xác định được giá trị góc γ_k giữa robot và cột mốc.

3.1.3.3. *Khảo sát sai số định góc của cảm biến ảnh toàn phương.*

Sai số nói chung và sai số định góc của cảm biến ảnh toàn phương phụ thuộc nhiều vào điều kiện sáng của môi trường. Do camera nằm ở giữa tâm của robot nên để ước tính giá trị $var(\gamma_k)$ ta có thể lấy mẫu các phép đo bằng thực nghiệm như sau:

- Cột mốc màu đỏ hình trụ tròn rỗng, bên trong đặt đèn phát quang để hạn chế ảnh hưởng của điều kiện sáng môi trường.

- Đặt hướng 0° (sau khi trải ảnh toàn cảnh trên hình 3.13) của camera trùng với trục X_R (hình 3.2), hướng thẳng trục X_R thẳng góc với cột mốc màu đỏ.

- Dùng thước vuông góc, vẽ một đường thẳng (vạch phấn trắng) trùng với trục Y_R (hình 3.2).

- Cho robot quay đúng 90° (khi vạch trắng trùng với trục X_R của robot) với một số lần N hữu hạn, (ví dụ $N = 30$) và ta có được các giá trị đo góc γ^o của camera-omni tương ứng mỗi lần quay như trên bảng 3.2.

$$\text{Lúc này giá trị } var(\gamma_k) = \frac{1}{N} \sum_{i=1}^N \left[(\gamma_i - \bar{\gamma}) \frac{\pi}{180} \right]^2 = 2,9920 \times 10^{-4} \text{ rad}^2$$

trong đó $\bar{\gamma}$ là giá trị trung bình của mẫu đo được,

$$\bar{\gamma} = \sum_{i=1}^N (\gamma_i / N) = 90,13^\circ$$

Bảng 3.2. Các mẫu phép đo của cảm biến camera toàn phương.

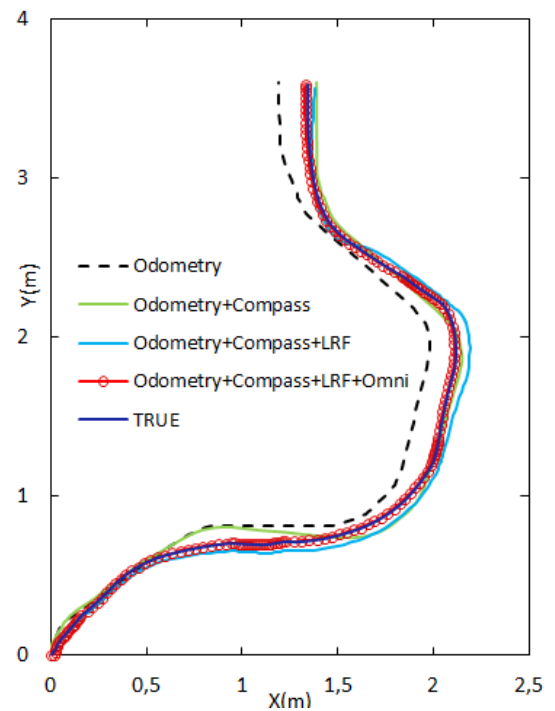
N	γ° (độ)	N	γ° (độ)	N	γ° (độ)
1	89	11	89	21	89
2	91	12	90	22	91
3	89	13	89	23	89
4	91	14	91	24	91
5	91	15	89	25	91
6	89	16	91	26	89
7	91	17	89	27	91
8	91	18	90	28	91
9	89	19	89	29	91
10	90	20	91	30	91

3.1.3.4. Kết quả thực nghiệm và thảo luận.

Môi trường thí nghiệm trong trường hợp này cũng như ở phần trên, nhưng phần này chúng tôi thêm vào một cột mốc màu đỏ hình trụ tròn có đường kính $d_{\text{mốc}} = 10$ cm với tọa độ biết trước. Bằng camera toàn phương, có thể xác định được góc hướng giữa trục robot và cột mốc.



(a)

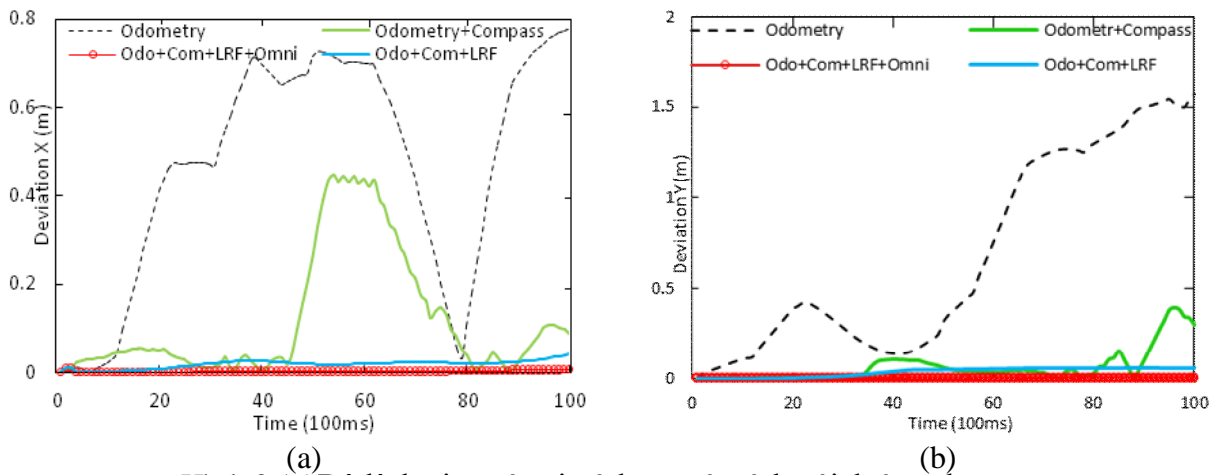


(b)

Hình 3.14 a) Ảnh chụp liên tiếp robot chuyển động trong môi trường; b) Quỹ đạo ước tính của robot với các cấu hình EKF khác nhau.

Hình 3.14a là chuỗi ảnh chụp liên tiếp robot chuyển động trong môi trường. Hình 3.14b là kết quả đường đi thực của robot (màu xanh đen) và các đường đi được ước lượng do các phép tổng hợp cảm biến khác với cảm biến odometry. Đường màu đen nét đứt là đường ước lượng với chỉ 1 cảm biến odometry. Đường màu xanh lá cây là đường ước lượng của phép tổng hợp giữa 2 cảm biến chỉ hướng và odometry. Đường màu xanh dương là đường ước lượng của phép tổng hợp giữa 3 cảm biến chỉ hướng, đo xa LRF và odometry. Cuối cùng, đường màu đỏ là đường ước lượng của phép tổng hợp của cả 4 cảm biến.

Hình 3.15 biểu diễn trực quan đồ thị các độ lệch trong các trường hợp tổng hợp cảm biến khác nhau so với đường thực. Từ các số liệu các kết quả thu được bằng thực nghiệm, ta có thể xác định được sai số căn quân phương RMSE (*root-mean-square error*) giữa đường đi của robot được ước tính qua EKF với đường đi thực như trên bảng 3.3



Hình 3.15 Độ lệch giữa các vị trí được ước tính với đường thực..
a) Theo phương X; b) Theo phương Y.

Bảng 3.3. Sai số căn quân phương RMSE của các cấu hình tổng hợp cảm biến so với đường thực.

Cấu hình	RMSE theo phương X(m)	RMSE theo phương Y(m)
Odometry	0,554	0,951
Odometry + Compass	0,190	0,224
Odometry + Compass+LRF	0,030	0,040
Odometry+Compass+LRF+Omni	0,021	0,038

Kết quả thực nghiệm trên hình 3.14b và bảng 3.3 cho thấy hiệu quả của bộ lọc Kalman dùng cho tổng hợp đa cảm biến: với càng nhiều cảm biến thì kết quả ước tính vị trí càng gần với các giá trị đường thực hơn.

3.2. Xây dựng bản đồ dẫn đường bằng tổng hợp dữ liệu cảm biến đo xa laser.

Trong nhiều trường hợp, lập bản đồ là một khâu quan trọng sau quá trình định vị. Nó cho phép robot căn cứ vào đó để vạch ra đường đi tới đích. Có nhiều phương pháp lập bản đồ. Ở đây tác giả đề xuất một phương pháp sử dụng kỹ thuật tổng hợp dữ liệu từ cảm biến laser để trích xuất thông tin bản đồ dẫn đường 2D từ đám mây các số liệu điểm ảnh 3D thu thập được. So với các bản đồ 2D thông thường, bản đồ này chứa đựng những thông tin xác đáng hơn do những đặc điểm 3D phong phú mang lại. Quá trình xây dựng bản đồ được thực hiện qua 2 giai đoạn: thu thập số liệu 3D và tổng hợp dữ liệu cảm biến theo không gian để có được bản đồ 2D.

3.2.1. Thu thập dữ liệu 3D các điểm ảnh của vật cản.

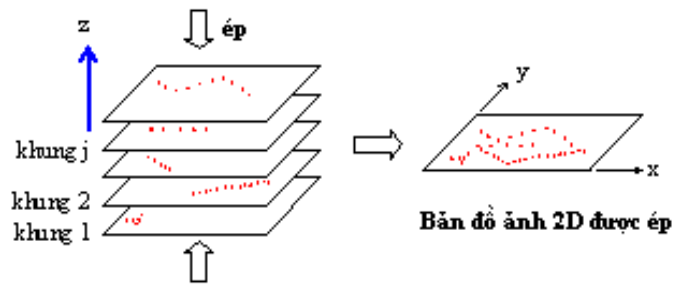
Như mô tả ở chương 2, để nhận được hình ảnh 3D của môi trường, máy đo xa laser được gắn lên một đế có thể quay ngẩng lên-xuống quanh một trục nằm ngang như mô tả chi tiết ở trên. Như vậy, một ảnh 3D thu được bao gồm j khung ảnh 2D phân bố đều theo chiều dọc với các góc ngẩng α_j . Mỗi khung ảnh 2D này là một ảnh được tạo bởi quá trình quét ngang của tia laser với tập các góc quét β_k . Mỗi điểm ảnh trong một khung được trả về một cặp giá trị khoảng cách R_k và β_k . Từ đó có thể xác định được tọa độ Descartes của mỗi điểm ảnh (j,k) tạo nên hình ảnh 3D của bề mặt các vật đối diện với robot:

$$\begin{aligned}x_{j,k} &= R_k \cos \alpha_j \cos \beta_k \\y_{j,k} &= R_k \cos \alpha_j \sin \beta_k \\z_{j,k} &= R_k \sin \alpha_j\end{aligned}\tag{3.39}$$

3.2.2. Giải thuật IPaBD xây dựng bản đồ dẫn đường 2D.

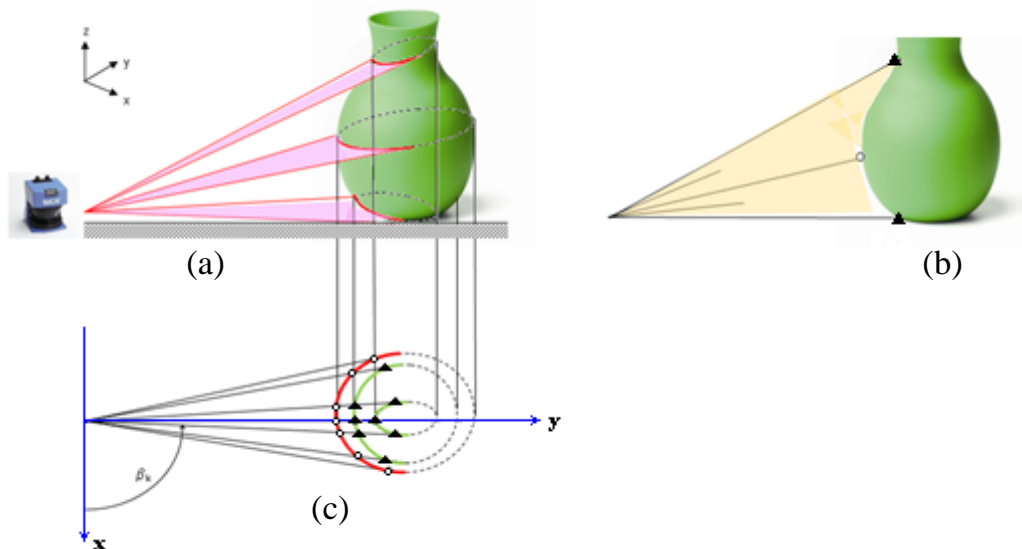
Yêu cầu đặt ra là phải xây dựng được bản đồ dẫn đường robot đến đích an toàn trên mặt phẳng 2D (OXY) khi có được dữ liệu các điểm ảnh 3D của các vật cản có kích thước cả về chiều rộng cũng như chiều cao.

Do robot di chuyển trên mặt phẳng sàn nhà, nên có thể sử dụng hình chiếu điểm ảnh các vật lên mặt phẳng (OXY) song song với sàn nhà để xây dựng bản đồ 2D này. Như trên vừa nói, toàn bộ dữ liệu thu được từ cảm biến laser trong một quá trình quét ngang lên-xuống là tập các dữ liệu điểm ảnh trên các khung ảnh. Các dữ liệu khung lần lượt được thu thập nối tiếp nhau theo chiều dọc với mỗi bước tăng góc ngang α_j . Mỗi khung là kết quả của quá trình thu thập dữ liệu theo mặt ngang (quét ngang) hết 13,3 ms với các góc quét lần lượt là β_k . Chú ý rằng thời gian quét ngang là rất ngắn (13,3 ms) và thời gian quét ngang cũng không quá dài (vài giây) so với quá trình biến động của các vật thể trong môi trường. Do đó có thể coi tập hợp dữ liệu các điểm ảnh 3D nhận được từ một cảm biến laser được thu thập chỉ liên quan đến j không gian khác nhau liền kề. Vậy ta hoàn toàn có thể tổng hợp các dữ liệu cảm biến này để trích xuất ra thông tin xác đáng xây dựng bản đồ dẫn đường. Tác giả đã đề xuất một giải thuật mới sử dụng phương pháp tổng hợp dữ liệu một cảm biến theo không gian, cho phép xây dựng bản đồ dẫn đường 2D từ các đám mây điểm ảnh 3D. Giải thuật được đặt tên là giải thuật “*nén ảnh và phát hiện vật cản*” IPaBD (Image Pressure and Barrier Detection). Quy trình thực hiện như sau. Trước tiên, thực hiện phép hợp (U) tất cả các điểm ảnh có tọa độ (x,y), ta được kết quả “ép” các khung trong ảnh 3D của môi trường theo phương z thành một bản đồ 2D duy nhất trên mặt phẳng OXY như hình 3.16.



Hình 3.16 Ép ảnh 3D thành 2D trên mặt phẳng OXY.

Ảnh bản đồ 2D thu được như trên là một ma trận các điểm ảnh, mỗi điểm được biểu diễn bằng một cặp thông số $(\beta_{j,k}, R_{j,k})$, trong đó $\beta_{j,k}$ là góc quét của tia laser thứ k tại góc ngắm j, và $R_{j,k}$ là khoảng cách đo được đến điểm ảnh. Sự phụ thuộc giữa hai thông số này là không đơn trị như trong mỗi khung quét laser 2D thông thường. Tức là, ứng với một góc quét ngang β_k có thể có một hoặc nhiều điểm có R khác nhau (do ứng với các độ cao z khác nhau) như hình chiếu trên ví dụ minh họa trong hình 3.17(b) chỉ ra.



Hình 3.17 a) Ví dụ minh họa quét ảnh 3D; b) hình chiếu các điểm ảnh hướng về camera trên mặt phẳng xy, ứng với một giá trị góc quét β_k có nhiều điểm ảnh (tròn hay tam giác) có các giá trị R khác nhau; c) mặt cắt đứng của quá trình quét với các điểm ảnh cùng có giá trị quét β_k nhưng có chiều cao khác nhau nên có giá trị R khác nhau.

Tuy nhiên, nhận thấy bản đồ 2D kẻ trên qua công đoạn ép ảnh thực ra là hình ảnh bề mặt hướng về phía robot của vật. Do đó với mục đích chỉ muốn đảm bảo tránh vật cản cho robot trong quá trình di chuyển, bản đồ 2D này chỉ cần chứa đựng các điểm thuộc biên vùng bề mặt trước “tầm nhìn” (thị trường) của robot. Đó là các điểm có khoảng cách đến cảm biến là cực tiểu trong số các điểm có cùng góc quét β_k , là các điểm được ký hiệu hình chấm tròn trên hình 3.17b. Kết quả như ví dụ hình 3.17b, ta chỉ còn lại một hình vòng cung màu đỏ trên bản đồ biểu diễn biên của vật gần nhất với robot mà thôi (hình 3.17c). Các điểm ảnh còn lại trên cùng góc quét (hình chấm tam giác) có các cao độ z khác, nhưng có khoảng cách đến robot xa hơn, thì được loại bỏ.

Sau bước xử lý này, sự phụ thuộc (β, R) trở nên đơn trị và có thể dễ dàng áp dụng các thuật toán phân đoạn ảnh cho bản đồ và tiến hành các xử lý vạch đường tiếp theo.

Cuối cùng, do robot chỉ có chiều cao giới hạn nên có thể đi qua an toàn dưới các điểm có độ cao lớn hơn một mức ngưỡng nào đó. Do đó không cần quan tâm, phải giữ lại các điểm ảnh được chọn ở bước trên nhưng lại có chiều cao z lớn hơn mức ngưỡng đó. Cũng vậy, các điểm ảnh có giá trị $z = 0$ tương ứng với mặt sàn cũng được loại bỏ khỏi cơ sở dữ liệu của bản đồ.

Tóm lại, toàn bộ các bước xây dựng bản đồ dẫn đường 2D bằng phương pháp tổng hợp dữ liệu cảm biến laser được tiến hành lần lượt theo các bước của giải thuật IPaBD như sau:

Giải thuật 3D-to-2D IPaBD

- Bước 1: Xuất phát từ tập hợp số liệu điểm ảnh 3D, thực hiện phép hợp tất cả các điểm ảnh lên mặt phẳng tọa độ (x,y) .
 - Bước 2: Với mỗi góc quét ngang β_k , tìm và chọn 1 giá trị $R_k = R_{min}$.
 - Bước 3: Loại bỏ các điểm có $z > z_{ngưỡng}$ và $z = 0$.
 - Bước 4: Thực hiện giải thuật phân đoạn ảnh (nếu cần).
-

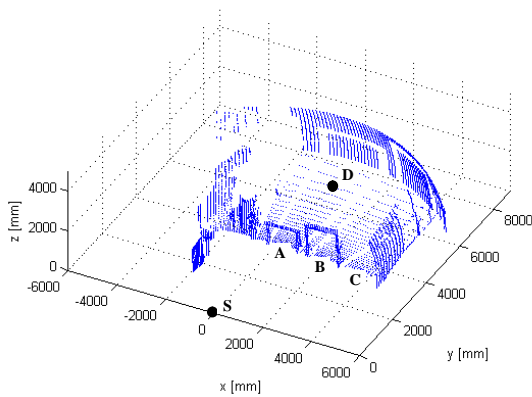
3.2.3. Thực nghiệm xây dựng bản đồ và áp dụng cho vạch đường đi và điều khiển robot.

Bản đồ dẫn đường đã được xây dựng bằng giải thuật IPaBD và đã được kiểm chứng bằng việc tiến hành các khâu vạch đường và điều khiển chuyển động robot trên đó. Trong nghiên cứu này, chúng tôi đã điều chỉnh các thông số của hệ đo xa laser sao cho với mỗi lần quay ngẩng lên (pitching up), cảm biến thu nhận được 94 khung ảnh 2D của môi trường, ứng với từng góc ngẩng trong dải từ -5° đến $+20^\circ$, với bước tăng góc ngẩng là $(5^\circ+20^\circ)/94 = 0,266^\circ$. Mỗi khung ảnh 2D nhận được ứng với 1 góc ngẩng α_j là do thu thập số liệu khi tia laser được quét ngang với các góc quét β_k trong dải quét 100° (từ 40° đến 140°), độ phân giải 1° . Như vậy, một ảnh 3D thu được sẽ là một đám mây với $101 \times 94 = 9.494$ điểm ảnh. Ảnh này có được tính tuyến tính và độ chính xác theo chiều cao là do tốc độ quét ngẩng được đảm bảo là không đổi. Do cảm biến được đặt ở độ cao so với mặt sàn $0,4\text{m}$ nên ở góc ngẩng cực tiểu -5° , cảm biến chỉ có thể phát hiện ra một vật có chiều cao toàn bộ từ 0 m trở lên ở khoảng cách xa đến $4,58\text{ m}$.

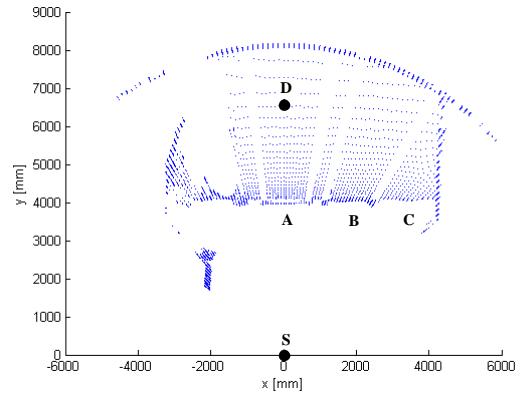
3.2.3.1. Kết quả xây dựng bản đồ bằng giải thuật IPaBD.

Môi trường thực nghiệm được thực hiện tại tiền sảnh nhà G2-trường Đại học Công nghệ. Với các thông số của thiết bị được chọn như trình bày trên, ảnh 3D thu được tại hiện trường như hình 3.18 với điểm xuất phát $S(0,0)$ mm và đích $D(0,6800)$ mm. Robot có chiều cao là 1200 mm . Tại hiện trường, có một cổng A có thanh dầm nằm thấp hơn chiều cao của robot và một cổng B có thanh dầm nằm cao hơn robot. Ngoài ra có một hành lang C với chiều cao không hạn chế nhưng ở vị trí bên cạnh. Robot cần chọn cho mình một đường đi từ S tới đích D ngắn nhất mà không va chạm với các vật cản.

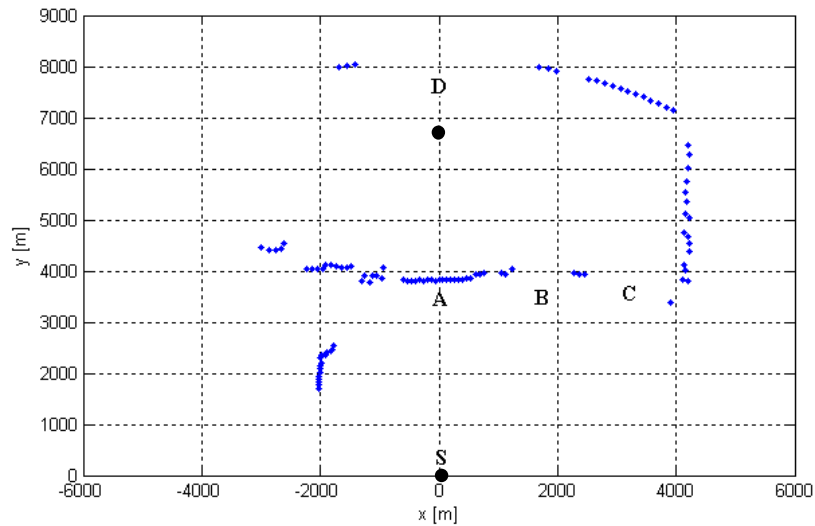
Hình 3.19 là ảnh này sau khi được "ép" lên mặt phẳng OXY với tất cả 9.494 điểm tạo ra bản đồ ảnh 2D.



Hình 3.18 Ảnh chụp 3D môi trường toàn cục.



Hình 3.19 Bản đồ 2D với toàn bộ điểm ảnh 3D được ép trên mặt phẳng OXY.



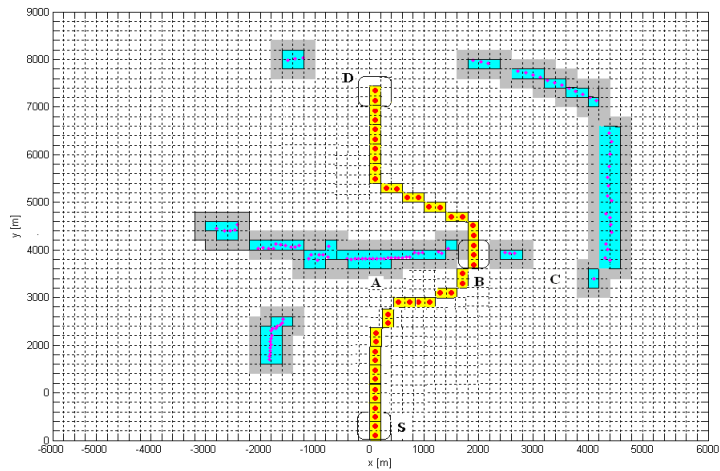
Hình 3.20 Kết quả bản đồ thu được do giải thuật IPaBD.

Hình 3.20 là bản đồ dẫn đường 2D chỉ còn chứa đựng các điểm cản phía trước tầm nhìn của robot được phát hiện qua giải thuật IPaBD. Bản đồ này cho thấy lượng dữ liệu được tinh giản đi rất nhiều, nhưng lại cho thấy rất rõ thông tin về không gian bị cản (chiếm giữ) như cổng A (có thanh dầm thấp hơn chiều cao robot) cùng các mặt cản thông thường khác; cũng như không gian tự do thuộc về các cổng B và hành lang C. Những kết quả này có được là nhờ có cơ sở dữ liệu 3D cùng giải thuật lập bản đồ theo phương pháp tổng hợp cảm biến laser. Đó là điều không phải thiết bị 2D thông thường nào cũng làm được.

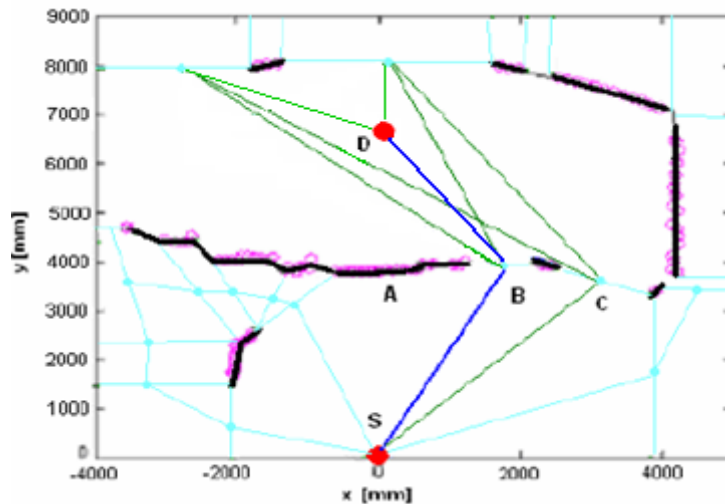
3.2.3.2. Thực nghiệm vạch đường đi và điều khiển robot nhờ bản đồ dẫn đường.

Chúng tôi đã sử dụng phương pháp PDBS với giải thuật đơn giản dựa trên khoảng cách Euclidean [102] cho quá trình phân đoạn ảnh trên bản đồ nhận được thành các đoạn thẳng. Bản đồ này tiếp tục được số hóa như sau. Không gian mặt sàn hoạt động của robot được rời rạc hóa thành một ma trận $M(j,k)$ ô chữ nhật, mỗi ô có kích thước $(a \times a)$ cm, gọi là ô chiếm giữ. Giá trị a được chọn bằng $1/3$ đường kính thiết diện ngang của robot. Như vậy tâm một ô chiếm giữ $M(j,k)$ sẽ có tọa độ là $(j.a+a/2, k.a+a/2)$ và chiếm một vùng tọa độ x từ $j.a$ đến $j(a+1)$ và vùng tọa độ y từ $k.a$ đến $k(a+1)$. Nếu điểm ảnh trên bản đồ có tọa độ nằm trong vùng tọa độ ô nào thì ô đó được gọi là bị chiếm giữ và được gán giá trị logic “1”, các ô còn lại thuộc về không gian tự do, có giá trị “0”. Bản đồ lưới này thực chất là một ma trận chứa các phần tử 0 và 1 trong đó các ô 1 được sử dụng để thể hiện địa hình có vật cản, robot không thể đi qua và ngược lại. Từ số liệu các vùng có các ô bị chiếm giữ, cần dùng giải thuật “dãn ảnh” (dilation) để dãn ra một ô nữa, tạo vùng không gian ngăn robot va chạm với mép vùng bị chiếm giữ. Từ các dữ liệu bản đồ các ô bị chiếm giữ bởi điểm ảnh các vật cản, cùng các giá trị của 2 ô xuất phát $S(x_S, y_S)$ và ô đích $D(x_D, y_D)$ có thể áp dụng giải thuật như giải thuật A^* [57] để vạch ra đường đi ngắn nhất qua các ô tự do để đến ô đích.

Kết quả lập trình sử dụng giải thuật tìm kiếm A^* từ điểm xuất phát S tới điểm đích D trên bản đồ số hóa từ hình 3.20 cho đường đi tối ưu qua các lưới chiếm giữ được biểu diễn trên hình 3.21. Ở đây không gian bản đồ được chia thành 40×40 ô. Do kích thước của robot là (60×60) cm, mỗi ô được chọn là (20×20) cm. Ta thấy bản đồ của các vùng bị chiếm giữ bao gồm cả phần dãn ảnh (có độ sáng đậm). Kết quả tìm đường cho thấy, robot không đi qua cổng A do phát hiện có vật cản là thanh dầm (do không gian 3 chiều) và cũng không đi qua hành lang C vì sẽ có đường đi dài hơn mà sẽ đi qua cổng B mặc dù cũng có thanh dầm nhưng lại nằm ở độ cao hơn chiều cao của robot.



Hình 3.21 Vạch đường cho robot trên bản đồ IPaBD bằng giải thuật tìm kiếm A*.



Hình 3.22 Bản đồ Voronoi với đường đi tối ưu qua nút B (đường đậm nét).

Cũng từ bản đồ hình 3.20, chúng tôi đã kiểm chứng khả năng ứng dụng của nó khi thực nghiệm lập trình một kiểu vạch đường đi khác theo kỹ thuật đồ thị Voronoi [29] với tất cả các đường đi qua các nút là điểm giữa các khoảng không gian tự do cho robot có thể di chuyển. Từ đó có thể chọn ra một đường đi ngắn nhất là đường S-B-D như hình 3.22 chỉ ra.

Quá trình điều khiển robot trong trường hợp này đơn giản được thực hiện như sau: từ điểm xuất phát, robot quay tại chỗ cho tới đúng hướng SB, sau đó chuyển động thẳng về B. Đến B, robot quay tại chỗ cho tới đúng hướng BD rồi chuyển động thẳng về D.

3.3. Kết luận.

Trong chương này tác giả đã trình bày quá trình nghiên cứu áp dụng phương pháp tổng hợp cảm biến với EKF nhằm nâng cao độ chính xác ước lượng vị trí của robot trong quá trình dẫn đường. Các quá trình tổng hợp cảm biến đã được tiến hành thực nghiệm cho kết quả tốt hơn (gần với vị trí thực hơn) so với trường hợp không dùng bộ lọc. Đó là quá trình tổng hợp dữ liệu cảm biến giá trị vị trí được ước tính từ cảm biến lập mã trục quay theo phương pháp odometry với các dữ liệu đo nhận được từ các cảm biến ngoài như cảm biến chỉ hướng từ-địa bàn, cảm biến đo xa laser và cảm biến ảnh toàn phương. Tác giả lần đầu tiên đã thử nghiệm việc tổng hợp cảm biến với 4 cảm biến như vậy trên một mô hình robot đa cảm biến ở Việt nam, cho được kết quả tốt, phù hợp giữa lý thuyết và thực nghiệm.

Phương pháp tổng hợp dữ liệu cảm biến theo không gian cũng đã được tác giả nghiên cứu để phát triển xây dựng bản đồ dẫn đường 2D từ các dữ liệu 3D phong phú. Qua đó, đã phát triển một giải thuật mới giải quyết vấn đề này với tên gọi *giải thuật ép ảnh và phát hiện vật cản IPaBD*. Các kết quả thử nghiệm ứng dụng các kỹ thuật phân đoạn ảnh bản đồ, kỹ thuật tìm đường đi tối ưu A*, kỹ thuật vạch đường theo đồ thị Voronoi đã được tác giả áp dụng thành công trên bản đồ này chứng tỏ khả năng ứng dụng của phương pháp xây dựng bản đồ được thực hiện. Nội dung liên quan đến chương này đã được công bố tại Danh mục các công trình khoa học của tác giả liên quan đến Luận án [3-9].

CHƯƠNG 4

ĐIỀU KHIỂN CHUYỂN ĐỘNG

Khâu *điều khiển chuyển động* có nhiệm vụ điều khiển robot bám quỹ đạo tiến về đích an toàn. Nhằm đảm bảo chất lượng điều khiển, việc chọn luật điều khiển ổn định là quan trọng. Trong chương này, tác giả trình bày việc thực thi một giải pháp sử dụng luật điều khiển ổn định theo tiêu chuẩn Lyapunov khi có mặt các can nhiễu cho robot di động hai bánh vi sai được chế tạo tại phòng thí nghiệm. Tiếp đó, một bộ lọc Kalman có các giá trị ước lượng vị trí tốt hơn thông thường được thiết kế xen vào vòng điều khiển phản hồi đã được mô phỏng và thử nghiệm. Kết quả cho phép nâng cao chất lượng điều khiển chuyển động.

Dưới đây là tóm tắt một số khái niệm cơ bản liên quan đến sự ổn định Lyapunov.

4.1. Sự ổn định Lyapunov và nguyên lý bất biến LaSalle [60]

4.1.1. Khái niệm ổn định Lyapunov đối với hệ tự trị

Xét một hệ thống tự trị (autonomous):

$$\dot{x} = f(x) \quad (4.1)$$

Trong đó $f : D \rightarrow R^n$ là một hàm Lipschitz cục bộ từ một miền $D \subset R^n$ vào R^n . Một điểm $x_c \in D$ được gọi là *điểm cân bằng* nếu $f(x_c) = 0$.

Để đơn giản không mất tính tổng quát, ta chỉ xét các khái niệm và tính chất của điểm cân bằng là gốc tọa độ $x = 0$.

Nếu hệ có điểm cân bằng khác 0 có thể dùng phép biến đổi để đưa điểm cân bằng về gốc:

Xét phép $y = x - x_c$ và đặt $g(y) = f(y - x_c)$

Khi đó $\dot{y} = \dot{x} = f(x) = f(y + x_c)$, $g(0) = 0 \quad \forall t$. Điểm cân bằng của hệ mới như vậy là $y = 0$.

Định nghĩa 4.1.1 *Điểm cân bằng $x = 0$ của (4.1) được gọi là*

(i) *Ổn định, nếu với mỗi $\varepsilon > 0$, tồn tại $\delta = \delta(\varepsilon) > 0$ sao cho*

$$\|x(0)\| < \delta \Rightarrow \|x(t)\| < \varepsilon, \quad \forall t \geq 0$$

(ii) Không ổn định, nếu nó không phải điểm ổn định.

(iii) Ổn định tiệm cận, nếu nó ổn định và δ có thể được chọn sao cho

$$\|x(0)\| < \delta \Rightarrow \lim_{x \rightarrow \infty} x(t) = 0.$$

Giả sử tồn tại hàm $V : D \rightarrow R$ là hàm khả vi liên tục và xác định dương,

$\dot{V}(x) = \frac{dV}{dt}$ là đạo hàm của $V(x)$ dọc theo quỹ đạo (4.1), nếu $\dot{V}(x) \leq 0$ thì $V(x)$ giảm

dọc theo các nghiệm của (4.1). Ta có các định lý ổn định sau:

Định lý 4.1.1 Cho $x = 0$ là điểm cân bằng của (4.1) và $D \subset R^n$ là một miền chứa $x = 0$. Giả sử $V : D \rightarrow R$ là hàm khả vi liên tục sao cho:

$$\begin{aligned} V(0) = 0, \quad V(x) > 0 \text{ trong } D - \{0\} \\ \dot{V}(x) \leq 0 \text{ trong } D. \end{aligned} \quad (4.2)$$

Khi đó $x = 0$ là ổn định. Hơn nữa, nếu

$$\dot{V}(x) < 0 \text{ trong } D - \{0\} \quad (4.3)$$

thì $x = 0$ là ổn định tiệm cận.

Hàm khả vi liên tục $V(x)$ thỏa (4.2) được gọi là hàm Lyapunov. Nói cách khác hệ ổn định tiệm cận tại $x = 0$ nếu có hàm Lyapunov.

Định lý 4.1.2 Cho $x = 0$ là điểm cân bằng của (4.1). Giả sử $V : R^n \rightarrow R$ là hàm khả vi liên tục sao cho

$$\begin{aligned} V(0) = 0, \quad V(x) > 0 \quad x \neq 0, \\ \|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty, \\ \dot{V}(x) < 0 \quad x \neq 0. \end{aligned} \quad (4.4)$$

Khi đó $x = 0$ là ổn định tiệm cận toàn cục.

4.1.2. Nguyên lý bất biến

Cho $x(t)$ là nghiệm của (4.1)

- Một điểm P được gọi là điểm giới hạn dương của $x(t)$ nếu có một dãy $\{t_n\}$ với $t_n \rightarrow \infty$ sao cho $x(t_n) \rightarrow P$ khi $n \rightarrow \infty$.

- Tập hợp tất cả các điểm giới hạn dương của $x(t)$ được gọi là tập giới hạn

duong của $x(t)$

- Một tập M được gọi là *tập bất biến* đối với (4.1) nếu

$$x(0) \in M \Rightarrow x(t) \in M, \quad \forall t \in R.$$

- Một tập M được gọi là *tập bất biến dương* đối với (4.1) nếu

$$x(0) \in M \Rightarrow x(t) \in M, \quad \forall t \geq 0.$$

Ta cũng nói rằng $x(t)$ tiếp cận tập M khi $t \rightarrow \infty$ nếu với mỗi $\varepsilon > 0$, có một $T > 0$ sao cho

$$d(x(t), M) < \varepsilon, \quad \forall t > T.$$

Trong đó $d(P, M)$ ký hiệu khoảng cách từ một điểm P đến tập M :

$$d(P, M) = \inf_{x \in M} \|P - x\|.$$

Điểm cân bằng ổn định tiệm cận là tập giới hạn dương của mọi nghiệm bắt đầu đủ gần điểm cân bằng. *Đường bao giới hạn ổn định* (stable limit cycle) là tập giới hạn dương của mọi nghiệm bắt đầu đủ gần đường bao giới hạn. Nghiệm sẽ tiến tới đường bao giới hạn khi $t \rightarrow \infty$.

Điểm cân bằng và đường bao giới hạn là các tập bất biến vì bất kỳ nghiệm nào xuất phát từ một tập sẽ ở lại trong tập đó $\forall t \in R$. Tập $\Omega_c = \{x \in R^n \mid V(x) \leq c\}$ với $\dot{V}(x) \leq 0$ đối với mọi $x \in \Omega_c$ là một tập bất biến dương bởi vì mọi nghiệm xuất phát từ Ω_c sẽ ở lại trong Ω_c với mọi $t \geq 0$.

Bổ đề 4.1.2 *Nếu một nghiệm của (4.1) bị chặn và thuộc D với $t \geq 0$, khi đó tập giới hạn dương của nó L^+ là một tập rỗng, compact và bất biến. Hơn nữa, $x(t) \rightarrow L^+$ khi $t \rightarrow \infty$*

Định lý 4.1.2 (Định lý bất biến LaSalle) *Cho $\Omega \subset D$ là tập compact và bất biến dương đối với (4.1). Cho $V : D \rightarrow R$ là hàm khả vi liên tục sao cho $\dot{V}(x) \leq 0$ trên Ω . Cho E là tập tất cả các điểm trong Ω sao cho $\dot{V}(x) = 0$. Gọi M là tập bất biến lớn nhất trong E . Khi đó mỗi nghiệm xuất phát trong Ω đều tiếp cận M khi $t \rightarrow \infty$.*

4.2. Xây dựng bộ điều khiển chuyển động ổn định.

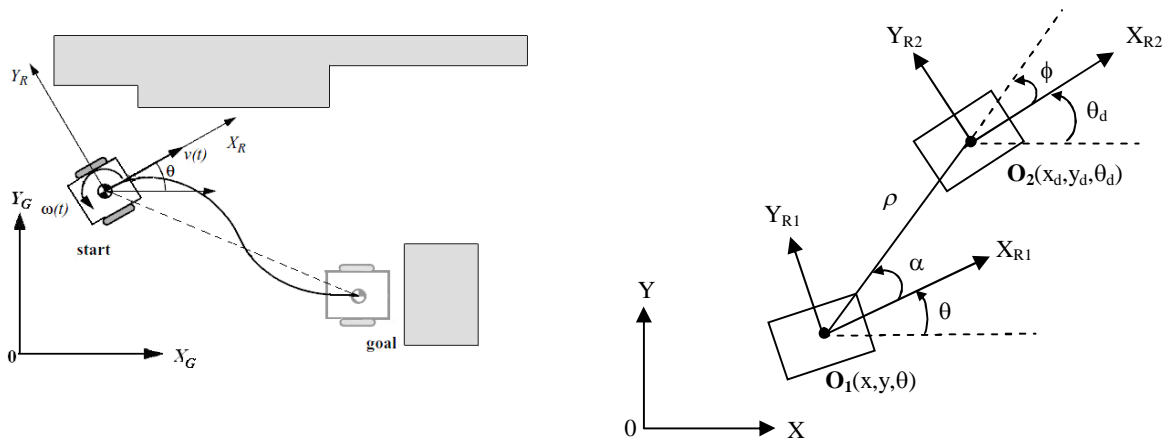
Để thực hiện nhiệm vụ điều khiển robot đi theo quỹ đạo đã được xác định trước thì phương pháp đơn giản hay được sử dụng là chia quỹ đạo thành tập hợp các

điểm. Khi đó nhiệm vụ trên sẽ trở thành bài toán điều khiển chuyển động điểm – điểm ổn định: đi từ điểm ban đầu tới điểm lân cận (đích phụ) và tiếp tục cho tới khi tới điểm đích.

Robot di động hai bánh vi sai có mô hình động học như biểu thức (3.1) là hệ thống phi tuyến với ràng buộc không khả tích (non-holonomic) sẽ gặp khó khăn trong vấn đề thiết kế luật điều khiển chuyển động phản hồi trạng thái tĩnh, trơn và bất biến với thời gian vì không thỏa mãn điều kiện của Brockett [25]. Một số nghiên cứu đã giải quyết vấn đề này bằng việc sử dụng phương pháp chuyển đổi mô hình động học sang hệ tọa độ cực với các biến dẫn đường, từ đó đưa ra các luật điều khiển phản hồi trơn để điều khiển robot chuyển động ổn định từ một vị trí bất kỳ (tọa độ và hướng của robot) tới vị trí đích như các tác giả Aicardi [17], Secchi [105],... Tuy nhiên các nghiên cứu đó chỉ xét với trường hợp lý tưởng khi mô hình robot không bị ảnh hưởng bởi nhiễu, trong khi thực tế cho thấy rằng luật điều khiển sẽ không còn ổn định tiệm cận về đích khi có nhiễu nhất là đối với đầu vào là vận tốc góc. Các tác giả [126] đã đề cập tới vấn đề này khi giải quyết bài toán có nhiễu cho một xe tự hành ba bánh. Theo đó, để giải quyết cho mô hình robot di động hai bánh vi sai được chế tạo với điều kiện có nhiễu, chúng tôi cũng chia các tập cấu hình hoạt động của robot (tọa độ và hướng) thành 2 miền: các cấu hình gần vị trí đích (x_d, y_d, θ_d) được gọi là tập cấu hình cục bộ và cấu hình xa vị trí đích gọi là tập cấu hình toàn cục. Luật điều khiển tương ứng với hai cấu hình đó được lựa chọn.

4.2.1. Mô hình động học.

Hình 4.1 mô tả tư thế robot khi nó được điều khiển chuyển động qua 2 vị trí tham chiếu trong hệ tọa độ toàn cục OXY. Gắn với robot là các hệ tọa độ cục bộ $Ox_R Y_R$. Robot xuất phát từ vị trí tùy ý O_1 , nơi nó có tọa độ (x, y) và góc hướng θ . Robot cần được điều khiển đi tới vị trí đích O_2 nơi nó cần có tọa độ (x_d, y_d) và góc hướng θ_d đã biết.



Hình 4.1 Tư thế và các thông số của robot.

Hệ phương trình động học của robot được biểu diễn như sau:

$$\begin{aligned}\dot{x} &= v \cos \theta \\ \dot{y} &= v \sin \theta \\ \dot{\theta} &= \omega\end{aligned}\quad (4.5)$$

Trong đó, v và ω lần lượt là vận tốc dài và vận tốc góc của robot. Gọi O_1 và O_2 là các *điểm cân bằng* của hệ (4.5).

Luật điều khiển theo tiêu chuẩn ổn định Lyapunov có thể nhận được khi chuyển đổi các biến cấu hình (x, y, θ) thành các biến dẫn đường (ρ, ϕ, α) . Với ρ là khoảng cách từ O_1 đến O_2 , ϕ là góc tạo bởi hai hướng O_1O_2 và O_2X_{R2} , $\alpha = (\phi + \theta_d) - \theta$ là góc tạo bởi hai hướng O_1O_2 và O_1X_{R1} .

Nếu α nằm trong khoảng $\alpha \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, ta có:

$$\begin{aligned}\rho &= \sqrt{(x_d - x)^2 + (y_d - y)^2} \\ \phi &= \text{atan2}(y_d - y, x_d - x) - \theta_d \\ \alpha &= \text{atan2}(y_d - y, x_d - x) - \theta\end{aligned}\quad (4.6)$$

Mô hình động học của robot lúc này được mô tả qua các biến dẫn đường:

$$\begin{aligned}
\dot{\rho} &= -v \cos \alpha \\
\dot{\phi} &= \frac{v \sin \alpha}{\rho} \\
\alpha &= -\omega + \frac{v \sin \alpha}{\rho}
\end{aligned} \tag{4.7}$$

Trong trường hợp α nằm trong khoảng còn lại $\alpha \in \left(-\pi, -\frac{\pi}{2}\right] \cup \left(\frac{\pi}{2}, \pi\right]$, có thể định nghĩa lại hướng tiến của robot bằng việc đặt $v = -v$.

Khi tính đến nhiễu, gọi ε_x , ε_y , ε_θ là các nhiễu đo ảnh hưởng tới các giá trị danh định của tọa độ (x, y) và góc hướng θ . Các giá trị ước lượng vị trí đo được phản hồi về bộ điều khiển sẽ là: $\hat{x} = x + \varepsilon_x$, $\hat{y} = y + \varepsilon_y$ và $\hat{\theta} = \theta + \varepsilon_\theta$. Trong đó $|\varepsilon_x| \leq \varepsilon_x^{\max}$, $|\varepsilon_y| \leq \varepsilon_y^{\max}$, $|\varepsilon_\theta| \leq \varepsilon_\theta^{\max}$ bị chặn, và ε_x^{\max} , ε_y^{\max} , $\varepsilon_\theta^{\max}$ lần lượt là các giá trị chặn trên của ε_x , ε_y , ε_θ .

Các biến dẫn đường ρ , ϕ , α cũng chịu ảnh hưởng của các nhiễu phản hồi trạng thái ε_ρ , ε_ϕ , ε_α như sau:

$$\begin{aligned}
\varepsilon_\rho &= \sqrt{(x_d - \hat{x})^2 + (y_d - \hat{y})^2} - \sqrt{(x_d - x)^2 + (y_d - y)^2} \\
\varepsilon_\phi &= a \tan 2(x_d - \hat{x})^2 + (y_d - \hat{y})^2 - a \tan 2(x_d - x)^2 + (y_d - y)^2 \\
\varepsilon_\alpha &= \varepsilon_\phi - \varepsilon_\theta
\end{aligned} \tag{4.8}$$

Trong đó: $|\varepsilon_\rho| \leq \varepsilon_\rho^{\max}$, $|\varepsilon_\phi| \leq \varepsilon_\phi^{\max}$, $|\varepsilon_\alpha| \leq \varepsilon_\alpha^{\max}$, với ε_ρ^{\max} , ε_ϕ^{\max} , $\varepsilon_\alpha^{\max}$ lần lượt là các giá trị chặn trên của ε_ρ , ε_ϕ , ε_α .

Tương tự, gọi $|\varepsilon_v| \leq \varepsilon_v^{\max}$, $|\varepsilon_\omega| \leq \varepsilon_\omega^{\max}$ lần lượt là các nhiễu đầu vào của các tín hiệu điều khiển v và ω , với ε_v^{\max} , $\varepsilon_\omega^{\max}$ là các giá trị chặn trên của ε_v , ε_ω .

Mô hình (4.8) khi tính đến ảnh hưởng của nhiễu đầu vào sẽ trở thành:

$$\begin{aligned}
\dot{\rho} &= -(v + \varepsilon_v) \cos \alpha \\
\dot{\phi} &= (v + \varepsilon_v) \frac{\sin \alpha}{\rho} \\
\dot{\alpha} &= -(\omega + \varepsilon_\omega) + (v + \varepsilon_v) \frac{\sin \alpha}{\rho}
\end{aligned} \tag{4.9}$$

4.2.2. Thiết kế bộ điều khiển.

Đặt $\Omega = \{(x, y, \theta) : \rho, \alpha, \phi \in R\}$ là tập hợp của tất cả các cấu hình của robot trong không gian cấu hình.

Đặt $\Omega_L = \{(x, y, \theta) : \rho(x, y) < \varepsilon_p \cap |\phi(x, y) - \alpha(x, y, \theta)| < \varepsilon_\theta\}$ là tập hợp cấu hình cục bộ bao gồm tất cả các cấu hình nằm ở lân cận cấu hình đích. Trong đó $\varepsilon_p, \varepsilon_\theta$ được xem là các giá trị nhỏ.

Đặt $\Omega_G = \Omega - \Omega_L$ là tập cấu hình toàn cục bao gồm tất cả các cấu hình nằm ở xa cấu hình đích.

4.2.2.1. Điều khiển ổn định trong tập cấu hình toàn cục Ω_G .

Chọn hàm Lyapunov (là hàm xác định dương) được xây dựng trên các biến dẫn đường có dạng:

$$V = V_{G1} + V_{G2} = \frac{\rho^2}{2} + \frac{\alpha^2 + h\phi^2}{2} > 0 \tag{4.10}$$

Chú ý tới (4.7), đạo hàm bậc nhất của V là:

$$\dot{V} = \dot{V}_{G1} + \dot{V}_{G2} = \rho\dot{\rho} + (\alpha\dot{\alpha} + h\phi\dot{\phi}) = -\rho v \cos \alpha + \alpha \left[-\omega + v \cdot \frac{\sin \alpha}{\alpha} \cdot \frac{(\alpha + h\phi)}{\rho} \right]$$

Để đáp ứng điều kiện ổn định theo tiêu chuẩn Lyapunov thì đạo hàm bậc nhất của V phải có giá trị âm, Căn cứ vào đó, luật điều khiển cho v và ω được chọn như sau:

$$\begin{aligned}
v &= k_v \rho \cos \alpha \\
\omega &= k_\alpha \alpha + k_v \frac{\cos \alpha \sin \alpha}{\alpha} (\alpha + h\phi)
\end{aligned} \tag{4.11}$$

Trong đó các hệ số $k_v > 0, h > 0$ và $k_\alpha > 0$.

Luật điều khiển này trong điều kiện ảnh hưởng của *nhiều phản hồi trạng thái* ε_ρ và ε_α có dạng:

$$v = k_v(\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha)$$

$$\omega = k_\alpha(\alpha + \varepsilon_\alpha) + k_v \frac{\cos(\alpha + \varepsilon_\alpha) \sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \left[(\alpha + \varepsilon_\alpha) + h(\phi + \varepsilon_\phi) \right]$$

Thay $\dot{\rho}$ ở (4.10) vào (4.11), xét \dot{V}_{G1} :

$$\begin{aligned} \dot{V}_{G1} &= \rho \dot{\rho} = -\rho(v + \varepsilon_v) \cos \alpha \\ &= -\rho \cos \alpha \left[k_v(\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v \right] \\ &= -k_v \rho \cos \alpha (\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) - \rho \varepsilon_v \cos \alpha \\ &= -k_v \rho^2 \cos \alpha \cos(\alpha + \varepsilon_\alpha) - k_v \rho \varepsilon_\rho \cos \alpha \cos(\alpha + \varepsilon_\alpha) - \rho \varepsilon_v \cos \alpha \\ &= -k_v \rho \cos \alpha \cos(\alpha + \varepsilon_\alpha) (\rho + \varepsilon_\rho) - \rho \varepsilon_v \cos \alpha \end{aligned} \quad (4.12)$$

Xét các giá trị góc α và $\alpha + \varepsilon_\alpha$ nằm trong khoảng $\alpha, \alpha + \varepsilon_\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2} \right]$ nên các thành phần $\cos \alpha > 0$, $\cos(\alpha + \varepsilon_\alpha) > 0$. Trong tập cấu hình toàn cục Ω_G ta có $\rho > 0$, $\rho > |\varepsilon_\rho|$ nên $\rho + \varepsilon_\rho > 0$. Từ kết quả này ta có thể chọn hệ số k_v đủ lớn để bỏ qua can nhiễu đầu vào ε_v trong số hạng thứ 2 của (4.12). Do đó $\dot{V}_{G1} \leq 0$ và điều đó có nghĩa V_{G1} sẽ hội tụ về một giới hạn hữu hạn không âm, và ρ sẽ tiến tới một giá trị nhỏ.

Xét tiếp đối với \dot{V}_{G2}

$$\begin{aligned} \dot{V}_{G2} &= \alpha \dot{\alpha} + h\phi \dot{\phi} \\ &= \alpha \left[-(\omega + \varepsilon_\omega) + (v + \varepsilon_v) \frac{\sin \alpha}{\rho} \right] + h\phi (v + \varepsilon_v) \frac{\sin \alpha}{\rho} \end{aligned} \quad (4.13)$$

Đặt:

$$A = \alpha \left[-(\omega + \varepsilon_\omega) + (v + \varepsilon_v) \frac{\sin \alpha}{\rho} \right]; \quad B = h\phi (v + \varepsilon_v) \frac{\sin \alpha}{\rho}$$

Thay luật điều khiển (4.11) trong điều kiện ảnh hưởng của *nhiều phản hồi trạng thái* vào A và B:

$$\begin{aligned}
A &= \alpha \left\{ \left[k_v(\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v \right] \frac{\sin \alpha}{\rho} - (\omega + \varepsilon_\omega) \right\} \\
&= \alpha \left\{ k_v \cos(\alpha + \varepsilon_\alpha) \sin \alpha + k_v \varepsilon_\rho \cos(\alpha + \varepsilon_\alpha) \frac{\sin \alpha}{\rho} + \varepsilon_v \frac{\sin \alpha}{\rho} \right\} \\
&\quad - \alpha \left\{ k_\alpha (\alpha + \varepsilon_\alpha) + k_v \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha) [(\alpha + \varepsilon_\alpha) + h(\phi + \varepsilon_\phi)] + \varepsilon_\omega \right\} \quad (4.14a)
\end{aligned}$$

$$\begin{aligned}
&= -k_\alpha \alpha^2 - k_\alpha \alpha \varepsilon_\alpha - \alpha \varepsilon_\omega + k_v \alpha \cos(\alpha + \varepsilon_\alpha) [\sin \alpha - \sin(\alpha + \varepsilon_\alpha)] + k_v \alpha \varepsilon_\rho \cos(\alpha + \varepsilon_\alpha) \frac{\sin \alpha}{\rho} \\
&\quad + \alpha \varepsilon_v \frac{\sin \alpha}{\rho} - k_v h \alpha \phi \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha) - k_v h \alpha \varepsilon_\phi \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha)
\end{aligned}$$

$$\begin{aligned}
B &= h\phi(v + \varepsilon_v) \frac{\sin \alpha}{\rho} = h\phi \frac{\sin \alpha}{\rho} [k_v(\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v] \\
&= k_v h\phi \frac{\sin \alpha}{\rho} (\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + h\phi \varepsilon_v \frac{\sin \alpha}{\rho} \quad (4.14b) \\
&= k_v h\phi \sin \alpha \cos(\alpha + \varepsilon_\alpha) + k_v h\phi \varepsilon_\rho \frac{\sin \alpha}{\rho} \cos(\alpha + \varepsilon_\alpha) + h\phi \varepsilon_v \frac{\sin \alpha}{\rho}
\end{aligned}$$

$$\begin{aligned}
\dot{V}_{G_2} &= A + B = -k_\alpha \alpha^2 - k_\alpha \varepsilon_\alpha \alpha - \alpha \varepsilon_\omega + k_v \alpha \cos(\alpha + \varepsilon_\alpha) [\sin \alpha - \sin(\alpha + \varepsilon_\alpha)] \\
&\quad + k_v h\phi \cos(\alpha + \varepsilon_\alpha) \left[\sin \alpha - \frac{\alpha}{(\alpha + \varepsilon_\alpha)} \sin(\alpha + \varepsilon_\alpha) \right] + \varepsilon_v \frac{\sin \alpha}{\rho} (h\phi + \alpha) \quad (4.14c) \\
&\quad + k_v \varepsilon_\rho \left(\frac{\alpha}{\rho} + \frac{h\phi}{\rho} \right) \sin \alpha \cos(\alpha + \varepsilon_\alpha) - k_v h \alpha \varepsilon_\phi \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha)
\end{aligned}$$

Vì ε_α nhỏ nên 2 biểu thức trong các ngoặc vuông có thể được tính gần đúng khi chú ý rằng $\cos \varepsilon_\alpha \approx 1$ và $\sin \varepsilon_\alpha \approx \varepsilon_\alpha$:

$$[\sin \alpha - \sin(\alpha + \varepsilon_\alpha)] \approx -\varepsilon_\alpha \cos \alpha$$

$$\left[\sin \alpha - \frac{\alpha}{\alpha + \varepsilon_\alpha} \sin(\alpha + \varepsilon_\alpha) \right] \approx \varepsilon_\alpha \left(\frac{\sin \alpha - \alpha \cos \alpha}{\alpha + \varepsilon_\alpha} \right)$$

Do đó (4.14c) có thể viết lại thành:

$$\begin{aligned}
\dot{V}_{G_2} = A + B \approx & -k_\alpha \alpha^2 - k_\alpha \varepsilon_\alpha \alpha - \alpha \varepsilon_\omega - k_v \alpha \varepsilon_\alpha \cos(\alpha + \varepsilon_\alpha) \cos \alpha \\
& + k_v h \phi \varepsilon_\alpha \cos(\alpha + \varepsilon_\alpha) \left(\frac{\sin \alpha - \alpha \cos \alpha}{\alpha + \varepsilon_\alpha} \right) + \varepsilon_v \frac{\sin \alpha}{\rho} (h \phi + \alpha) \\
& + k_v \varepsilon_\rho \left(\frac{\alpha}{\rho} + \frac{h \phi}{\rho} \right) \sin \alpha \cos(\alpha + \varepsilon_\alpha) - k_v h \alpha \varepsilon_\phi \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha)
\end{aligned} \tag{4.14d}$$

Từ (4.14d) ta có thể chọn hệ số k_α đủ lớn để bỏ qua nhiều $\varepsilon_\alpha, \varepsilon_v, \varepsilon_\rho, \varepsilon_\phi$ (các thành phần phía sau) để $\dot{V}_{G_2} \leq 0$ trong cấu hình toàn cục Ω_G . Do đó V_{G_2} sẽ hội tụ về một giới hạn hữu hạn không âm và α sẽ tiến tới một giá trị nhỏ.

Như vậy với luật điều khiển đã chọn ở (4.11) thì $\dot{V}_G = \dot{V}_{G_1} + \dot{V}_{G_2} \leq 0$ là hàm bán xác định âm, hàm Lyapunov V_G là hàm dương. Hệ thống sẽ xuất phát từ trong cấu hình toàn cục Ω_G tiến đến trong cấu hình cục bộ Ω_L .

Phương trình hệ thống (4.9) với luật điều khiển (4.11) có nhiều sẽ trở thành:

$$\begin{aligned}
\dot{\rho} &= - \left[k_v (\rho + \varepsilon_\rho) \cos \alpha \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v \cos \alpha \right] \\
\dot{\phi} &= \left[k_v (\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v \right] \frac{\sin \alpha}{\rho} \\
\dot{\alpha} &= \left[k_v (\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) + \varepsilon_v \right] \frac{\sin \alpha}{\rho} \\
&- \left\{ k_\alpha (\alpha + \varepsilon_\alpha) + k_v \frac{\sin(\alpha + \varepsilon_\alpha)}{(\alpha + \varepsilon_\alpha)} \cos(\alpha + \varepsilon_\alpha) \left[(\alpha + \varepsilon_\alpha) + h(\phi + \varepsilon_\phi) \right] + \varepsilon_\omega \right\}
\end{aligned} \tag{4.15}$$

4.2.2.2. Điều khiển ổn định hệ thống trong tập cấu hình cục bộ Ω_L .

Luật điều khiển (4.11) trong cấu hình toàn cục Ω_G như đã xét ở trên tuy vậy sẽ không ổn định trong cấu hình cục bộ Ω_L , việc này có thể chứng minh như sau.

Giả sử rằng ρ tiến tới giá trị nhỏ ε_ρ , (ε_ρ giống như ρ , luôn dương) các biến (α, ϕ) lần lượt tiến đến các giá trị xấp xỉ nhiều của chúng là ε_α và ε_ϕ . Xấp xỉ phương trình động học (4.15) gần cấu hình đích trở thành:

$$\dot{\rho} = - \left[k_v (\varepsilon_\rho + \varepsilon_\rho) + \varepsilon_v \right] \tag{4.16a}$$

$$\dot{\phi} = k_v \varepsilon_\alpha \left(1 + \frac{\varepsilon_\rho}{\varepsilon_\rho} \right) \tag{4.16b}$$

$$\dot{\alpha} = k_v \varepsilon_\alpha \left(1 + \frac{\varepsilon_\rho}{\varepsilon_p}\right) - 2\varepsilon_\alpha (k_\alpha + k_v) - 2k_v h \varepsilon_\phi - \varepsilon_\omega \quad (4.16c)$$

Xét hàm Lyapunov và luật điều khiển vẫn được giữ nguyên như trong cấu hình toàn cục Ω_G . Thay $\dot{\rho}$ ở (4.16a) vào (4.10) và xét \dot{V}_{G1}

$$\dot{V}_{G1} = \rho \dot{\rho} = k_v (-\varepsilon_p^2 - \varepsilon_p \varepsilon_\rho) - \varepsilon_p \varepsilon_v \quad (4.17)$$

Bằng cách chọn $\varepsilon_p \geq |\varepsilon_\rho| + \frac{|\varepsilon_v|}{k_v}$ để $\dot{V}_{G1} \leq 0$ tại biên giữa hai tập cấu hình toàn cục và tập cấu hình cục bộ. Khi ρ tiến tới giá trị nhỏ ε_p thì hệ thống bắt đầu tiếp cận vùng biên và vào vùng cục bộ.

Trong (4.17), với $k_v > 0$ như đã chọn trước, $V_{G1} = \frac{\rho^2}{2}$ bị chặn do đó ρ cũng bị chặn. Thành ra luật điều khiển của v vẫn đúng trong cấu hình cục bộ.

$$\text{Đặt: } \dot{V}_{G2} = \alpha \dot{\alpha} + h \phi \dot{\phi} = \dot{V}_{G2'} + \dot{V}_{G2''} \quad (4.18)$$

$$\text{Xét } \dot{V}_{G2'} = h \phi \dot{\phi} = h k_v \varepsilon_\phi \varepsilon_\alpha \left(1 + \frac{\varepsilon_\rho}{\varepsilon_p}\right) \quad (4.19)$$

Để thấy rằng trong (4.16b) có một finite-escape-time (tồn tại điểm thời gian hữu hạn t_1 mà tại đó lại có $\phi(t_1) = \infty$) khi $\varepsilon_\alpha \left(1 + \frac{\varepsilon_\rho}{\varepsilon_p}\right) > 0$, và (4.16c) cũng có cùng điều kiện giống như (4.16b), do α tỷ lệ với ϕ .

Khi $\varepsilon_\alpha, \varepsilon_\phi > 0$ thì $\dot{V}_{G2'} > 0$, ϕ không thể tiến về không, tức là hệ thống sẽ không ổn định.

Do đó ta sẽ phải thiết kế lại bộ điều khiển để hệ vòng kín ổn định bền vững.

Với $\alpha = (\phi + \theta_d) - \theta$, đặt $\theta_e = \theta - \theta_d$, hay từ (4.6) ta có $\theta_e = \phi - \alpha$

Hàm Lyapunov trong cấu hình cục bộ được lựa chọn khác như sau:

$$V_L = V_{L1} + V_{L2} = \frac{\rho^2}{2} + \frac{(\phi - \alpha)^2}{2} > 0 \quad (4.20a)$$

Hay

$$V_L = V_{L1} + V_{L2} = \frac{\rho^2}{2} + \frac{\theta_e^2}{2} > 0 \quad (4.20b)$$

Luật điều khiển ω trong cấu hình cục bộ của Ω_L được lựa chọn lại như sau:

$$\begin{aligned} v &= k_v(\rho + \varepsilon_\rho) \cos(\alpha + \varepsilon_\alpha) \\ \omega &= -k_\theta \theta_e \end{aligned} \quad (4.21)$$

Như đã chọn ở (4.17) (luật điều khiển v vẫn đúng cho cấu hình cục bộ) thì $\dot{V}_{L1} = \dot{V}_{G1} \leq 0$, và ρ, ϕ, α bị chặn.

Xét \dot{V}_{L2} ,

$$\begin{aligned} \dot{V}_{L2} &= \theta_e \dot{\theta}_e = \theta_e (\omega + \varepsilon_\omega) \\ &= \theta_e (-k_\theta \theta_e + \varepsilon_\omega) = -k_\theta \theta_e^2 + \theta_e \varepsilon_\omega \end{aligned} \quad (4.22)$$

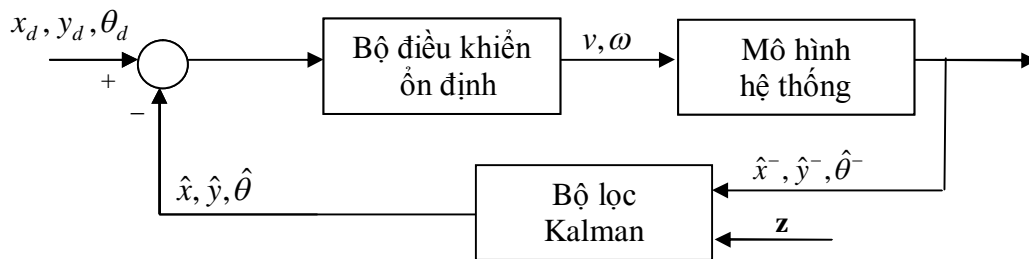
Từ (4.22) chọn k_θ đủ lớn để $\dot{V}_{L2} \leq 0$ hay $\dot{V}_L = \dot{V}_{L1} + \dot{V}_{L2} \leq 0$. Ở cấu hình toàn cục với luật điều khiển (4.11) hệ thống sẽ xuất phát từ cấu hình toàn cục tiến vào cấu hình cục bộ, khi giá trị ρ tiến tới giá trị nhỏ ε_p thì hệ thống bắt đầu chuyển sang cấu hình cục bộ với luật điều khiển (4.21). Do $\dot{V}_L \leq 0$ bị chặn tại biên giữa hai tập cấu hình toàn cục Ω_G và tập cấu hình cục bộ, Ω_L nên $\Omega_L = \{x \in R^n \mid V_L \leq c\}$ bị chặn, V_L có giới hạn là $c > 0$ khi $t \rightarrow \infty$. Tập $\Omega_L = \{x \in R^n \mid V(x) \leq c\}$ với $\dot{V}_L \leq 0$ đối với mọi $x \in \Omega_L$ là một tập bất biến dương bởi vì mọi nghiệm xuất phát từ Ω_L sẽ ở lại trong Ω_L với mọi $t \geq 0$. Hay nói cách khác, mỗi quỹ đạo xuất phát trong Ω_L phải nằm lại trong Ω_L và hội tụ tiệm cận đến điểm cân bằng O_2 khi $t \rightarrow \infty$. Ω_L cũng được gọi là một ước lượng miền hấp thụ hay miền ổn định tiệm cận (region of attraction or region of asymptotic stability) [60]. Lúc đó $\rho \rightarrow 0$; $\theta_e \rightarrow 0$, hay $x \rightarrow x_d$; $y \rightarrow y_d$; $\theta \rightarrow \theta_d$

4.2.3. Sử dụng bộ lọc Kalman cho vòng điều khiển phản hồi.

Trong chương 3, luận án đã sử dụng kỹ thuật tổng hợp cảm biến bằng bộ lọc Kalman cho việc định vị chính xác robot di động hai bánh vi sai. Kết quả cho thấy

giá trị ước lượng vị trí nhận được từ EKF gần với giá trị danh định của robot hơn bình thường. Điều này tương tự như đã giảm thiểu ảnh hưởng của nhiễu đo.

Trong khâu điều khiển chuyển động, các nhiễu đầu vào ($\varepsilon_v, \varepsilon_\omega$) và nhiễu đo ($\varepsilon_x, \varepsilon_y, \varepsilon_\theta$) ảnh hưởng đáng kể đến hiệu quả của mô hình điều khiển như bám đúng quỹ đạo và hội tụ về miền đích. Luận án đã thử nghiệm thiết kế một vòng điều khiển phản hồi kín trong chương trình như sơ đồ hình 4.2 trong đó bộ lọc Kalman được xen vào đường phản hồi nhằm nâng cao độ tin cậy của ước lượng vị trí robot. Do các giá trị ước lượng trạng thái ở đầu ra khi có bộ lọc EKF tin cậy hơn khi không có, nên các giá trị sai lệch từ đầu ra bộ so sánh với giá trị đích cấp tới đầu vào bộ điều khiển ổn định sẽ có độ tin cậy cao hơn. Kết quả làm cho chất lượng điều khiển tốt hơn.



Hình 4.2. Vòng điều khiển phản hồi với bộ lọc Kalman.

Kết quả mô phỏng và thực nghiệm cho thấy khi sử dụng bộ lọc EKF trong vòng phản hồi của hệ thống điều khiển chuyển động cho phép tăng độ chính xác của giá trị phản hồi, do đó giảm sai lệch và kiểm soát được giới hạn của nhiễu phản hồi so với trường hợp thông thường.

4.2.4. Mô phỏng và thực nghiệm.

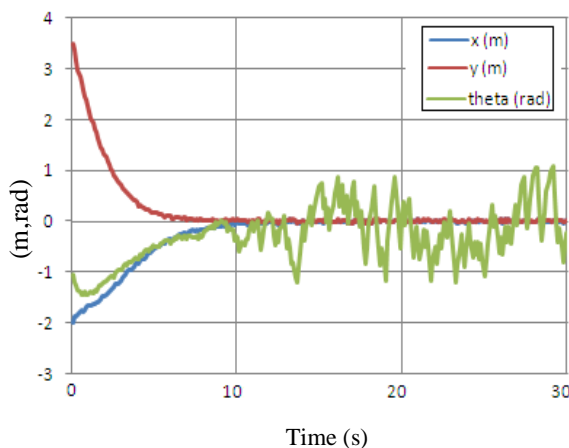
Mô phỏng được thực hiện trong MATLAB trong đó các tham số được trích xuất từ robot thực được xây dựng tại phòng thí nghiệm. Luật điều khiển thỏa mãn tiêu chuẩn ổn định Lyapunov trong cả hai tập cấu hình Ω_G và Ω_L đã được áp dụng.

Vận tốc cực đại của robot là 1.3m/s, thời gian lấy mẫu của hệ thống $\Delta t = 100$ ms. Sai số của hệ thống khi về đích là khoảng cách $\rho = 10^{-2}$ m. Giá trị tham số được chọn như sau: $k_v = 1$; $k_\alpha = 2$; $k_\theta = 1$ và $h = 5$. Các giá trị nhiễu đo được lựa chọn

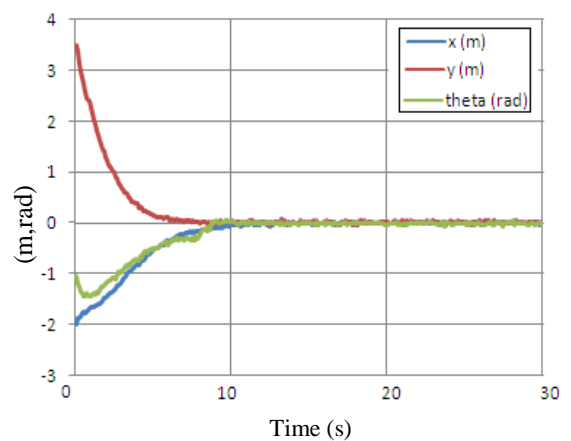
dựa trên cơ sở là sai lệch lớn nhất ở phần ước lượng của từ bộ EKF đối với robot thực là $\varepsilon_x^{max} = 0.1$ m, $\varepsilon_y^{max} = 0.1$ m, và $\varepsilon_\theta^{max} = 0.0036$. Nhiều hệ thống dựa trên khảo sát thực nghiệm với robot thực điều khiển motor bằng thuật toán PID với sai số của vận tốc góc ω_L và ω_R là $\pm 5\%$. Vì thế với $v_{max} = 1.3$ m/s thì $\varepsilon_v^{max} = 0.065$ và $\varepsilon_\omega^{max} = 0.2167$. Giá trị ε_p để chuyển sang cấu hình cục bộ được chọn sao cho thỏa

$$\text{mãn điều kiện } \varepsilon_p \geq \left| \varepsilon_\rho \right| + \frac{|\varepsilon_v|}{k_v}$$

- Trong mô phỏng 1 khi khảo sát tính ổn định, chúng tôi chọn robot có cấu hình đích là $(0, 0, 0^\circ)$, cấu hình xuất phát là $(-2, 3.5, -60^\circ)$. Robot được điều khiển chuyển động ổn định từ vị trí xuất phát đến vị trí đích. Kết quả thu được khi sử dụng chỉ một luật điều khiển (4.11) cho cả hai cấu hình nhận được ở hình 4.3a. Mặc dù tọa độ đường đi ổn định về đích sau 150 bước thời gian lấy mẫu với tọa độ $(x,y) = (0.0032$ m, 0.0013 m) nhưng góc hướng θ vẫn tồn tại khác không và thẳng giáng khá mạnh. Trong khi đó kết quả ở hình 4.3b cho thấy hiệu quả khi phân tách ra hai cấu hình với sử dụng luật (4.21) cho cấu hình cục bộ thì cả 3 biến đều đã ổn định tiệm cận về đích về không cho cả $(x,y,\theta) = (0.0086$ m, 0.0035 m, -0.0031 rad).



(a)



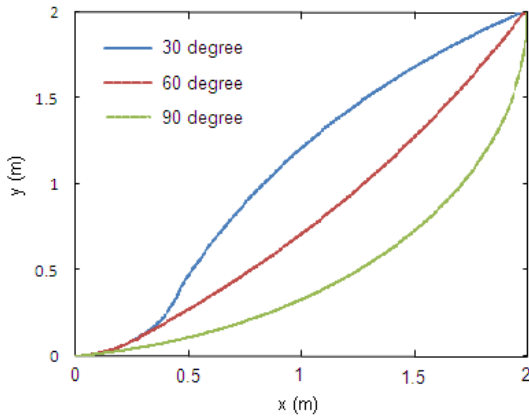
(b)

Hình 4.3. Đáp ứng với các luật điều khiển dùng 1 cấu hình (a) và 2 cấu hình (b).

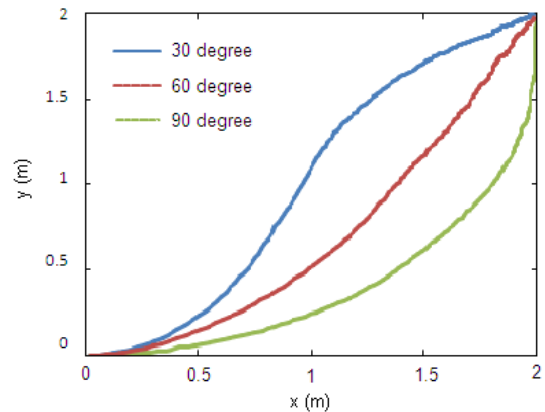
- Trong mô phỏng 2, vị trí xuất phát của robot là $(0, 0, 0^\circ)$ và 3 vị trí đích khác không là: $(2, 2, 30^\circ)$, $(2, 2, 60^\circ)$ và $(2, 2, 90^\circ)$. Hình 4.4 cho thấy kết quả mô

phòng trong đó các cấu hình đích của robot được hội tụ đến tọa độ (2, 2) với ba góc hướng khác nhau. Điều này cho thấy tính khả thi của bộ điều khiển.

• Thực nghiệm dẫn đường cũng được tiến hành trên robot hai bánh vi sai của phòng thí nghiệm từ vị trí xuất phát đến vị trí đích như vậy cho kết quả đạt yêu cầu như hình 4.5 gần như mô phỏng.



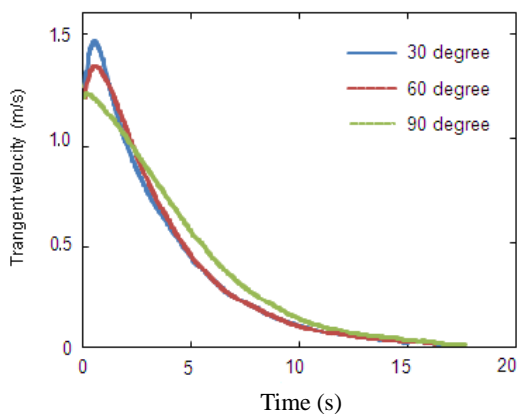
Hình 4.4 Kết quả mô phỏng.



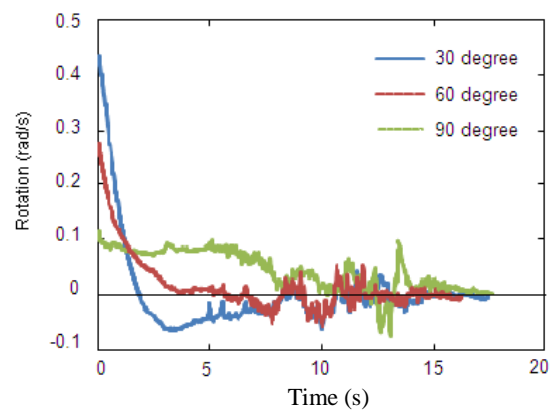
Hình 4.5 Kết quả thực nghiệm.

• Kết quả điều khiển ổn định của mô hình điều khiển bám quỹ đạo, hội tụ và ổn định ở miền đích được thử nghiệm trong 2 trường hợp:

- Khi không sử dụng bộ lọc Kalman: kết quả đo đạc trên hình 4.6a cho vận tốc dài v và hình 4.6b cho vận tốc góc ω .



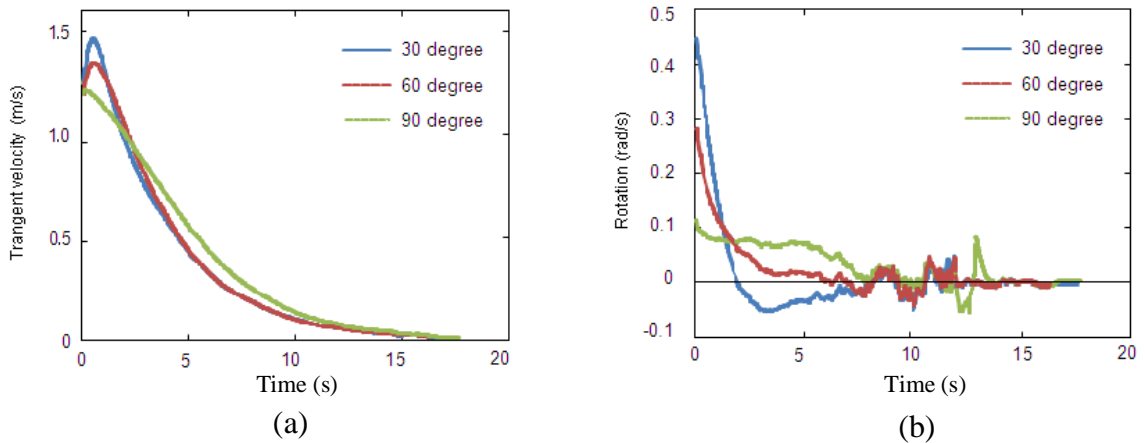
(a)



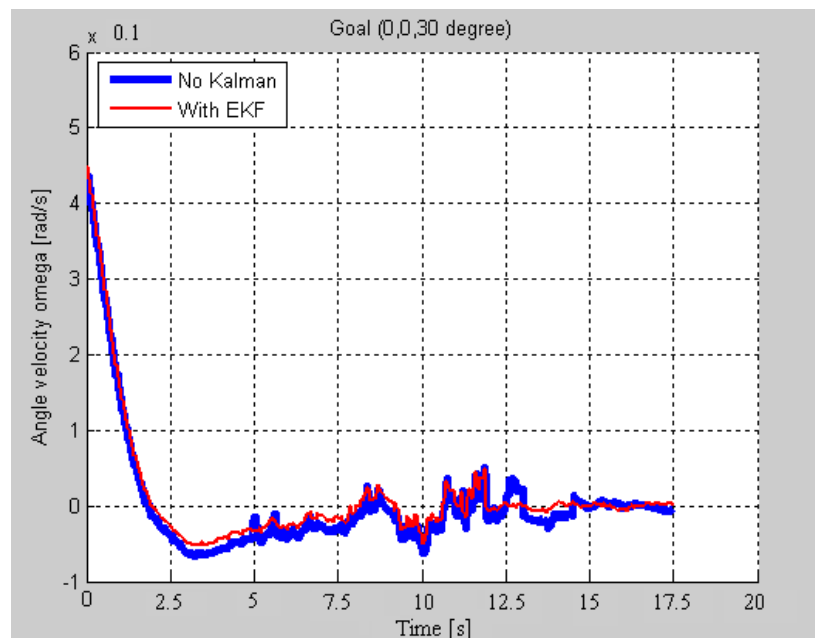
(b)

Hình 4.6. Kết quả điều khiển ổn định khi không có bộ lọc Kalman.

- Khi có sử dụng bộ lọc Kalman: kết quả đo đạc trên hình 4.7a cho vận tốc dài v và hình 4.7b cho vận tốc góc ω .



Hình 4.7. Kết quả điều khiển ổn định có bộ lọc Kalman.



Hình 4.8. Vận tốc góc ω tiệm cận ổn định đến đích khi không dùng bộ lọc EKF (đường màu xanh) và có dùng EKF (đường màu đỏ).

- Đồ thị hình 4.8 được trích xuất từ các đồ thị hình 4.6b và 4.7b là một so sánh trực quan sự biến thiên của vận tốc góc ω theo thời gian với trường hợp có và không có bộ lọc Kalman khi vị trí đích là $(0, 0, 30^\circ)$. Dễ thấy rằng, thông qua bộ lọc Kalman với sự tổng hợp dữ liệu cảm biến (đường màu đỏ) thì sự ước lượng trạng thái robot (đặc biệt là góc hướng) sẽ được tốt hơn, do đó vận tốc góc ω của robot

được điều khiển ở gần miền lân cận đích ổn định hơn so với trường hợp không có bộ lọc Kalman (đường màu xanh).

4.3. Kết luận.

Phần này trình bày những nghiên cứu thực thi quá trình điều khiển ổn định bám quỹ đạo cùng với việc sử dụng các giá trị định vị robot bằng kỹ thuật tổng hợp cảm biến với bộ lọc mở rộng Kalman EKF cho khâu điều khiển chuyển động. Kết quả việc kết hợp sử dụng các luật điều khiển chuyển động ổn định trong 2 tập cấu hình theo tiêu chuẩn Lyapunov cùng với EKF như vậy cho phép tăng tính chính xác và ổn định của quỹ đạo chuyển động. Phương pháp hàm Lyapunov tuy có thể đòi hỏi một quãng đường đi dài hơn nhưng bù lại sẽ có được một quỹ đạo liên tục, đáp ứng được cả góc hướng của robot tại điểm đích. Nội dung chương này đã được công bố tại Danh mục các công trình khoa học của tác giả liên quan đến Luận án [10].

KẾT LUẬN VÀ THẢO LUẬN HƯỚNG PHÁT TRIỂN

Robot di động tự trị là một đối tượng mới nên những năm gần đây nó đã được quan tâm bởi nhiều nhà nghiên cứu trên thế giới. Ở Việt Nam, trong công cuộc công nghiệp hóa và hiện đại hóa, đối tượng này cũng nhận được sự chú ý của các nhà khoa học thuộc lĩnh vực điện tử - điều khiển tự động. Trong số các vấn đề liên quan thì quá trình dẫn đường cho robot di động là quan trọng. Luận án này tập trung đi sâu nghiên cứu và đề ra một số giải pháp thực nghiệm cho một trong những phương pháp nâng cao chất lượng dẫn đường cho robot di động là *phương pháp tổng hợp cảm biến*. Việc sử dụng bộ lọc Kalman mở rộng để kết hợp những ưu nhược điểm của một hoặc nhiều cảm biến cho phép tăng được độ chính xác, tin cậy của việc định vị robot. Qua đó xây dựng được bản đồ và vạch đường đi mong muốn, cũng như điều khiển robot chuyển động ổn định tiệm cận đích.

1. Các nội dung được thực hiện trong Luận án.

Để đạt được mục tiêu kể trên, Luận án đã được tiến hành qua các bước sau. Đã nghiên cứu tổng quan lý thuyết liên quan đến kỹ thuật tổng hợp cảm biến áp dụng cho dẫn đường robot di động. Trên cơ sở nghiên cứu các đặc điểm vật lý của một số cảm biến hiện đại, nghiên cứu các mạch điều khiển điện tử, cơ khí cũng như kỹ thuật truyền thông; đã thiết kế chế tạo một mô hình robot đa cảm biến phục vụ cho các thử nghiệm dẫn đường thực tế. Nghiên cứu áp dụng thành công kỹ thuật tổng hợp cảm biến sử dụng đến 4 cảm biến (lập mã trục quay, chỉ hướng từ-địa bàn, đo xa laser và camera toàn phương) cho việc nâng cao độ tin cậy của phép định vị robot. Các dữ liệu nhận được trên một cảm biến đo xa laser trong không gian khác nhau cũng được tổng hợp theo một giải thuật đề xuất để xây dựng thử nghiệm bản đồ dẫn đường 2D chứa các thông tin phong phú từ các điểm ảnh 3D. Bản đồ này đã được kiểm chứng qua việc áp dụng thành công lên nó 2 kỹ thuật vạch đường đi là kỹ thuật tìm kiếm A* và đồ thị vạch đường Voronoi. Việc sử dụng các giá trị định vị tin cậy bằng bộ lọc EKF làm các biến tham chiếu cho khâu điều khiển chuyển động cũng đã được nghiên cứu thử nghiệm trong Luận án. Đó là việc chia quá trình

điều khiển chuyển động giữa 2 điểm khi có nhiễu của robot hai bánh vi sai thành 2 cấu hình với 2 luật điều khiển khác nhau, kết hợp với việc sử dụng bộ lọc Kalman trong vòng điều khiển phản hồi, cho phép cải thiện chất lượng điều khiển.

2. Những đóng góp chính của Luận án.

1. Đã thiết kế, xây dựng thành công một robot di động đa cảm biến, đặc biệt là đã cải tiến thành công một thiết bị đo xa laser 2D thành một cảm biến đo xa 3D. Hệ thống đã được đo kiểm, đảm bảo đủ tin cậy cho việc áp dụng thử nghiệm các quá trình điều khiển dẫn đường được thực hiện trong Luận án.

2. Đã thực nghiệm thành công quá trình tổng hợp tới 4 cảm biến hiện đại (lập mã trục quay, chỉ hướng từ-địa bàn, đo xa laser và camera toàn phương). Kết quả đạt được trên cơ sở nghiên cứu áp dụng phương pháp tổng hợp cảm biến với bộ lọc Kalman mở rộng đã nâng cao độ chính xác của phép định vị robot trong quá trình dẫn đường.

3. Đã nghiên cứu đề xuất một giải thuật tổng hợp dữ liệu cảm biến IPaBD, cho phép xây dựng thành công bản đồ dẫn đường 2D chứa đựng các thông tin hình ảnh 3D phong phú.

4. Đã thực thi một giải pháp tăng hiệu quả của khâu điều khiển chuyển động cho robot hai bánh vi sai bằng việc chia khâu này thành 2 quá trình với 2 hàm Lyapunov khác nhau và kết hợp với việc sử dụng kỹ thuật tổng hợp cảm biến bằng bộ lọc Kalman trong vòng điều khiển phản hồi.

3. Hướng phát triển tương lai.

- Do tốc độ truyền dữ liệu nối tiếp theo chuẩn RS 232C phổ biến mới chỉ hạn chế ở mức 38,400 baud nên thời gian quét 1 khung ảnh trong thiết bị đo xa laser còn dài. Vì vậy, tốc độ chuyển động của vật cần cần phát hiện trong các nghiên cứu này mới chỉ được giả thiết là không nhanh. Tác giả cùng nhóm nghiên cứu đã sử dụng công nghệ FPGA để tăng quá trình xử lý thu thập dữ liệu và cũng đạt được một số kết quả còn khiêm tốn. Tuy nhiên trong tương lai, có thể nghiên cứu chế tạo các phần cứng cho phép sử dụng tốc độ truyền lên đến 500.000 baud của thiết bị để rút

ngắn thời gian thu thập dữ liệu lại, từ đó cho phép phát hiện rộng rãi hơn các đối tượng chuyển động nhanh quanh robot.

- Tất cả kết quả nghiên cứu kể trên chỉ mới thực hiện ở môi trường trong nhà, hướng sắp tới đề tài sẽ áp dụng các kết quả trên cho việc dẫn đường các xe tự hành di chuyển trên các địa hình ngoài trời.

DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. T. T. Hoang, D. A. Viet, T. Q. Vinh (2011), “A 3D image capture system using a laser range finder”, *IEICE Proceeding of the 2th international conference on Integrated Circuit Design*, pp.76-81.
2. Trần Thuận Hoàng, Đặng Anh Việt và Trần Quang Vinh (2011), “Xây dựng hệ đo xa 3D sử dụng cảm biến laser dùng cho robot di động tự trị”, *Hội nghị toàn quốc về Điều khiển và Tự động hoá - VCCA-2011*, tr. 257-260.
3. Hoang T. T., Duong P. M., Van N. T. T., Viet D. A. and Vinh T. Q. (2012) “Development of a Multi-Sensor Perceptual System for Mobile Robot and EKF-based Localization”, *IEEE Proc. Conf. on Systems and Informatics*, pp. 519-522.
4. Hoang T. T., Viet D. A., Van N. T. T., Tuan P. D. and Vinh T. Q. (2012), “Extended Kalman Filter in Mobile Robot and FPGA-based Implementation”, *IEICE Proc. of the 3th Int. Conf. on Integrated Circuit Design*, pp. 167-172.
5. Trần Thuận Hoàng, Phùng Mạnh Dương, Đặng Anh Việt và Trần Quang Vinh (2012), “Robot di động đa cảm biến và định vị robot bằng phương pháp tổng hợp cảm biến với bộ lọc Kalman mở rộng”, *Hội thảo toàn quốc về Điện tử - Truyền thông – An toàn thông tin, ATC/REV*, tr. 130 – 135.
6. Tran Hiep Dinh, Manh Duong Phung, Thuan Hoang Tran, Quang Vinh Tran (2012), “Localization of a Unicycle-like Mobile Robot Using LRF and Omni-directional Camera”, *Proceedings 2012 IEEE International Conference on Control System, Computing and Engineering*, pp. 477-482.
7. T. T. Hoang, P. M. Duong, N. T. T. Van, D. A. Viet and T. Q. Vinh (2012), “Multi-Sensor Perceptual System for Mobile Robot and Sensor Fusion-based Localization”, *IEEE International Conference on Control, Automation and Informatics Sciences*, pp. 259-264
8. T. T. Hoang, P.M Duong, N.T.T.Van, D.A.Viet and T.Q. Vinh (2012), “Development of an EKF-based Localization Algorithm Using Compass Sensor and LRF”, *The 12th International Conference on Control, Automation, Robotics & Vision*, pp. 341-346.
9. Trần Thuận Hoàng, Phùng Mạnh Dương, Đặng Anh Việt và Trần Quang Vinh (2012), “Dẫn đường và tránh vật cản cho robot di động dựa trên ảnh laser 3D và siêu âm”, *Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 6*, tr. 451-458.

10. Tuan Hoang Tran, Manh Duong Phung, Thi Thanh Van Nguyen, Quang Vinh Tran (2012), “Stabilization Control of the Differential Mobile Robot Using Lyapunov Function And Extended Kalaman Filter”, *Vietnam Journal of Sciences and Technology* Vol. 50(4), pp. 441-452.

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. Phạm Hải An, Nguyễn Quang Hùng, Nguyễn Văn Chúc, Ngô Trọng Mai (2009), “Nâng cao tính khả dụng của hệ dẫn đường phương tiện cơ giới tích hợp MEMS-INS/GPS bằng hệ suy diễn nơron mờ”, *Tạp chí nghiên cứu khoa học và công nghệ quân sự, Viện Khoa học và công nghệ quân sự*, tr 5-11.
2. Phạm Hải An (2011), *Về một phương pháp nhận dạng chuyển động cho một lớp phương tiện cơ giới quân sự sử dụng đa cảm biến*, Luận án tiến sỹ kỹ thuật, Viện khoa học và công nghệ quân sự, Hà nội.
3. Lê Hoàng Anh, Ngô Văn Thuyên (2012), “Định vị cho Robot di động sử dụng camera và vật mốc”, *Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 6*, tr. 679-683.
4. Phạm Tuấn Hải (2012), *Nâng cao chất lượng hệ dẫn đường thiết bị bay trên cơ sở áp dụng phương pháp xử lý thông tin kết hợp*, Luận án tiến sỹ kỹ thuật, Viện khoa học và công nghệ quân sự, Hà nội.
5. Huỳnh Thái Hoàng (2012), “Thiết kế và thực hiện xe lăn điện điều khiển bằng mắt”, *Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 6*, tr. 529 – 535.
6. Vũ Bá Huân (2005), *Nghiên cứu, thiết kế và thử nghiệm hệ camera bám và theo dõi đối tượng chuyển động*, Luận văn thạc sỹ, Đại học Quốc gia Hà Nội, Hà Nội.
7. Lê Hùng Lân, Phạm Hải An (2005), “Khái niệm trộn dữ liệu đa cảm biến, áp dụng quy tắc Bayesian cho quá trình trộn”, *Tuyển tập báo cáo khoa học, Hội nghị khoa học kỹ thuật đo lường toàn quốc lần thứ 4*, tr 883-888.
8. Lê Hùng Lân, Nguyễn Quang Hùng, Phạm Hải An (2006), “Áp dụng Lý thuyết Dempster-Shafer cho quá trình trộn dữ liệu đa cảm biến”, *Tạp chí nghiên cứu khoa học và công nghệ quân sự, Trung tâm KHKT&CNQS*, tr. 164-170.

9. Lê Hùng Lân, Nguyễn Quang Hùng, Phạm Hải An (2008), “Tích hợp dữ liệu đa cảm biến trong đánh giá hướng chuyển động phương tiện giao thông mặt đất dựa trên các cảm biến quán tính sử dụng hệ chuyên gia mờ”, *Tạp chí nghiên cứu khoa học và công nghệ quân sự, Trung tâm KHKT&CNQS*, tr. 87-93.
10. Hà Khánh Linh, Trần Quang Vinh, Lê Vũ Hà (2004), “Ứng dụng thị giác máy tính trong điều khiển robot”, *Hội thảo Quốc gia- Một số vấn đề chọn lọc của Công nghệ thông tin*, tr. 229-238.
11. Trần Đức Tân, Huỳnh Hữu Tuệ (2006), “Định vị và dẫn đường trong thế kỷ 21”, *Tạp chí tự động hóa ngày nay* Tập 33(6), tr. 21-33.
12. Ngô Mạnh Tiến, Phan Xuân Minh, Phan Quốc Thắng, Nguyễn Doãn Phước (2012), “Một thuật toán điều khiển thích nghi theo mô hình mẫu cho robot tự hành non-holonomic với tham số bất định”, *Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 6*, tr. 607 – 613.
13. Nguyễn Văn Tính, Phạm Thượng Cát, Bùi Thị Thanh Quyên (2011), “Thiết kế quỹ đạo và điều khiển xe tự hành vận chuyển trong kho”, *Hội nghị toàn quốc về Điều khiển và Tự động hoá lần thứ 1*, tr.269-275.
14. Trần Quang Vinh (2003), *Nguyên lý phần cứng và kỹ thuật ghép nối máy tính*, NXB Giáo dục, Hà Nội.
15. Trần Quang Vinh (2005), “Xây dựng hệ thống cảm biến vật lý hỗn hợp dẫn đường cho robot di động”, *Báo cáo Hội nghị Vật lý toàn quốc lần thứ 4*, tr. 233-228.
16. Trần Quang Vinh, Vũ Tuấn Anh, Phùng Mạnh Dương và Trần Hiếu (2006), “Xây dựng robot di động được dẫn đường bằng các cảm biến siêu âm và cảm biến ảnh toàn phương”, *Tuyển tập công trình Hội nghị Cơ điện tử toàn quốc lần thứ 3*, tr.120-128.

Tiếng Anh

17. Aicardi M., Casalino G., Bicchi A. and Balestrino A. (1995), “Closed loop steering of unicycle-like vehicles via Lyapunov techniques”, *IEEE Robotíc & Automation Magazine* Vol. 2 (1), pp. 27-35.
18. Aufrere R., Gowdy J., Mertz C., Chuck Thorpe, Wang Chieh-Chih, Yata T. (2003), “Perception for collision avoidance and autonomous driving”, *Mechatronics* Vol. 13(10), pp. 1149 -1161.
19. Arras K. O., and Vestli S. J. (1998), “Hybrid, high-precision localization for the mail distributing mobile robot system POPS”, *Proceedings of the IEEE International Conference on Robotics and automation*, pp. 3129-3134.
20. Arras K. O. and Tomatis N. (1999), “Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision”, *Third European Workshop on Advanced Mobile Robots, IEEE*, pp. 177-185.
21. Byrne R. H., Klarer P. R., Pletta J. B. (1992), *Techniques for Autonomous Navigation*, Sandia Report SAND92-0457, Sandia National Laboratories, Albuquerque, New Mexico.
22. Borenstein J., Feng L. (1994), *A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots*, Technical Report UM-MEAM-94-22, University of Michigan, USA.
23. Baltzakis H., Antonis A., Trahanias P. (2003), “Fusion of laser and visual data for robot motion planning and collision avoidance”, *Journal of Machine Vision and Applications* Vol. 15(2), pp. 92-100.
24. Betke M., Gurvits L. (1997), “Mobile Robot Localization Using Landmarks”, *IEEE Transactions on Robotics and Automation, Mobile Robot Localization* Vol. 13(2), pp. 251-263.
25. Brockett R.W. (1983), “Asymptotic stability and feedback stabilization”, *Differential Geometric Control Theory*, Boston, Birkhauser, pp. 181-191.

26. Brooks R. R., Iyengar S. S. (1998), *Multi-Sensor Fusion: Fundamentals and Applications*, Prentice Hall PTR, New Jersey, USA.
27. Buchberger M., Jorg K.-W., von Puttkamer E. (1993), "Laserradar and sonar based world modeling and motion control for fast obstacle avoidance of the autonomous mobile robot MOBOT-IV", *Proceedings IEEE International Conference on Robotics and Automation*, pp. 534-540.
28. Bradski G., Kaehler A. (2008), *Learning OpenCV Library: Computer Vision with the OpenCV Library*, Nutshell Handbook, O'reilly media, inc., USA.
29. Bhattachrya P., Gavrilova M. L., (2008), "Roadmap-Based Path Planning Using the Voronoi Diagram for a Clearance-Based Shortest Path", *IEEE Robotics & Automation Magazine* Vol. 15(2), pp. 58-66.
30. Borenstein J., Koren Y. (1989), "Real-time obstacle avoidance for fact mobile robots", *IEEE Transactions on Systems, Man and Cybernetics* Vol. 19(5), pp. 1179-1187.
31. Borenstein J., Koren Y. (1991), "The vector field histogram - fast obstacle avoidance for mobile robots", *IEEE Journal of Robotics and Automation* Vol. 7(3), pp. 278-288.
32. Correa D. S. O., Sciotti D. F., Prado M. G., Sales D. O., Wolf D. F., Osório F. S. (2012), "Mobile Robots Navigation in Indoor Environments Using Kinect Sensor", *IEEE Second Brazilian Conference on Critical Embedded Systems*, pp. 35-41.
33. Cox I. J. (1991), "Blanche - An Experiment in Guidance and Navigation of an Autonomous Mobile Robot", *IEEE Transactions Robotics and Automation* Vol. 7(3), pp. 193-204.
34. Courtney J. D., Jain A. K. (1994), "Mobile Robot Localization via Classification of Multisensor Maps", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1672-1678.

35. Chenavier F., Crowley J. L. (1992), "Position Estimation for a Mobile Robot Using Vision and Odometry", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2588-2593.
36. Chung H., Ojeda L., Borenstein J. (2001), "Sensor fusion for mobile robot dead-reckoning with a precision-calibrated fiber optic gyroscope", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3588-3593.
37. Dong J., Zhuang D., Huang Y. and Fu J. (2009), "Advances in Multi-Sensor Data Fusion: Algorithms and Applications", *Sensors 2009* Vol. 9(10), pp. 7771-7784.
38. Dasarathy B. V. (1997), "Sensor fusion potential exploitation-innovative architectures and illustrative applications", *Proceedings of the IEEE* Vol. 85(1), pp. 24-38.
39. Dasarathy B. V. (2000), "More the merrier ... or is it? - sensor suite augmentation benefits assessment", *Proceedings of the 3rd International Conference on Information Fusion*, pp. 20-25.
40. Durrant-Whyte H., Stevens M. (2001), "Data fusion in decentralized sensing networks", *Information Fusion*, pp. 151-157.
41. Durrant-Whyte H. F. (1988), "Sensor models and multisensor integration", *International Journal of Robotics Research* Vol. 7(6), pp. 97-113.
42. Devantech Ltd. (n.d.), CMPS03 - Compass Module, Retrieved May 16, 2013, from <http://www.robot-electronics.co.uk/htm/cmeps3tech.htm>.
43. Devantech Ltd. (n.d.), SRF05 - Ultra-Sonic Ranger Technical Specification, Retrieved May 16, 2013, from <http://www.robot-electronics.co.uk/htm/srf05tech.htm>.
44. Duda R. O., Hart P. E. (1972), "Use of the Hough Transformation to Detect Lines and Curves in Pictures", *Communications of Association for Computing Machinery* Vol. 15(1), pp.11-15.

45. Deriche R., Vaillant R., Fauregas O. (1992), "From Noisy Edges Points to 3D Reconstruction of a Scene: A Robust Approach and Its Uncertainty Analysis", *World Scientific, Series in Machine Perception and Artificial Intelligence* Vol. 2, pp. 225 -232.
46. Evans J. M. (1994), "HelpMate: An autonomous mobile robot courier for hospitals", *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp.1695-1700.
47. Elmenreich Wilfried (2002), *An Introduction to Sensor Fusion*, Vienna University of Technology Research Report, Austria.
48. Elmenreich W., Pitzek S. (2001), "Using sensor fusion in a time-triggered network", *Proceedings of the 27th Annual Conference of the IEEE Industrial Electronics Society*, pp. 369-374.
49. Elfes A. (1989), "Using occupancy grids for mobile robot perception and navigation", *IEEE Computer* Vol. 22(6), pp. 46-57.
50. Feng L., Borenstein J., and Everett H. R. (1994), "Where am I?" *Sensors and Methods for Autonomous Mobile Robot Positioning*, Technical Report UM-MEAM-94-21, University of Michigan, USA.
51. Fabrizi E., Oriolo G., Panzieri S., Ulivi G. (2000), "Mobile robot localization via fusion of ultrasonic and inertial sensor data", *Proceedings of the 8th International Symposium on Robotics with Applications*, pp.71-78.
52. Grossmann A. and Poli R. (2001), "Robust mobile robot localization from sparse and noisy proximity readings using Hough transform and probability grids", *Robotics and Autonomous Systems* Vol. 37(1), pp. 1-18.
53. Garg D. P., and Kumar M. (2005), *Sensor modeling and multi-sensor data fusion*, Report number No. Rama-Aro-101, dept of Mechanical Engineering and Materials Science, Duke University, Durham.

54. Gan Q., Harris C. J. (2001), "Comparison of two measurement fusion methods for kalman-filter-based multisensor data fusion", *IEEE Transactions on Aerospace and Electronics* Vol. 37(1), pp.273-279.
55. Ha Khanh Linh (2005), Computer vision theory and applications for omnidirectional camera, a Thesis in Electronics and Telecommunication, Hanoi.
56. Harrion A., Newman P. (2008), "High Quality 3D Laser Ranging Under General Vehicle Motion", *IEEE International Conference on Robotics and Automation Pasadena*, pp. 7-12.
57. Hart P. E.; Nilsson N. J.; Raphael, B. (1968), "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics SSC4* Vol. 4(2), pp. 100–107.
58. Hahnel D., and Burgard W. (2002), "Map Building with Mobile Robots in Populated Environments", *Proceedings of the International conference on Intelligent Robots and Systems*, pp. 496-501.
59. Harrion A. and Newman P. (2008), "High Quality 3D Laser Ranging Under General Vehicle Motion", *IEEE International Conference on Robotics and Automation*, pp 7-12.
60. Hassan K. Khalil (1996), " Chapter 3- lyapunov Stability", *Nonlinear System*, Prentice Hall. Inc, New Jersey.
61. Hall David L. and Llinas J. (2001), *Handbook of multisensor data fusion*, Boca Raton, FL: CRC Press, Washington, D.C.
62. Hall DavidL, McMullen SonyaAH (2004), *Mathematical Techniques in Multisensor Data Fusion*, 2nd edition, Norwood, USA.
63. Hero A. O., Cochran D. (2011), "Sensor management: Past, Present, and Future", *IEEE Sensors Journal* Vol. 11(12), pp. 3064-3075.
64. Hoover A., Olsen B. D. (2000), "Sensor network perception for mobile robotics", *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 342- 347.

65. Iyengar S. S., Jayasimha D. N., Nadig D.. (1994), “A versatile architecture for the distributed sensor integration problem”, *IEEE Transaction on Computer* Vol. 43(2), pp. 175-185.
66. Jung B., Sukhatme G. S. (2001), “Tracking Multiple Moving Targets using a Camera and Laser Rangefinder”, *Institutue for Robotics and Intelligent Systems*, pp. 1-7.
67. Jensfelt P. (1999), *Localization using laser scanning and minimalistic environmental models*, Thesis, Royal Institute of Technology Stockholm, Sweden.
68. Julier S. J., Uhlmann J. K., Durrant-Whyte H. F. (1995), “A new approach for filtering nonlinear systems”, *Proceedings of the American Control Conference*, pp. 1628-1632.
69. Jensfelt P. (2001), *Approaches to Mobile Robot Localization in Indoor Environments*, Doctoral Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology Stockholm, Sweden.
70. Kumar S., Gupta D., Yadav S. (2010), “Sensor Fusion of Laser & Stereo Vision Camera for Depth Estimation and Obstacle Avoidance”, *International Journal of Computer Applications* Vol. 1(26), pp. 22-27.
71. Klein. L. A. (1999), *Sensor and Data Fusion Concepts and Applications*, SPIE Optical Engineering Press, Second Edition, Bellingham, WA, USA.
72. Kak A. C., Andress K. M., Lopez-Abadia C., Carroll M. S., Lewis J. R. (1990), “Hierarchical Evidence Accumulation in the Pseiki System and Experiments in Model-driven Mobile Robot Navigation”, *Proceedings of the Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-89)*, pp. 194-207.
73. Kalman R. E. (1960), “A new approach to linear fitltering and prediction problems”, *Transaction of the ASME, Journal of Basic Engineering*, pp. 35-45.

74. Koren Y., Borenstein J. (1991), "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation", *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 1398-1404.
75. Lawitzky G. (2001), "A navigation system for service robots: from research to products", *Proceedings of the 3rd International Conference on Field and Service Robotics*, pp. 15-19.
76. Leonard J., Durrant-Whyte H. F. (1991), "Mobile Robot Localization by Tracking Geometric Beacons", *IEEE Transactions on Robotics and Automation* Vol. 7(3), pp. 376-382.
77. Labayrade R., Gruyer Dominique, Cyril Royere, Mathias Perrollaz, Didier Aubert (2007), "Obstacle Detection Based on Fusion Between Stereovision and 2D Laser Scanner", in book, *Mobile Robots: Perception & Navigation*, pp. 91-110.
78. Lacroix S., Mallet A., Bonnafous D., Bauzil G., Fleury S., Herrb M., and Chatila R. (2002), "Autonomous Rover Navigation on Uneven Terrains: Functions and Integration", *International Journal of Robotics Research* Vol. 21(1), pp. 917-943.
79. Luo R. C., Kay M. (1989), "Multisensor integration and fusion in intelligent systems", *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 19(5), pp. 901-931.
80. Luo R. C., Su K. L. (1999), "A Review of High-Level Multisensor Fusion: Approaches and Applications", *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 25-31.
81. Makela H. (2001), "Overview of LHD navigation without artificial beacons", *Robotics and Autonomous Systems* Vol. 36 (1), pp. 21-35.
82. Moravec Hans P. (1988), "Sensor Fusion in Certainty Grids for Mobile Robots", *AI Magazine* Vol. 9(2), pp. 61-74.

83. Motion Mind: Motor Controller Data Sheet, Retrieved July 17, 2008, from [http:// www.solutions-cubed.com](http://www.solutions-cubed.com).
84. Miao Yu and Li Shu-qin (2010), “A Method of Robot Navigation Based on the Multi-Sensor Fusion”, *Intelligent Systems and Applications (ISA), 2nd International Workshop*, pp.1-4.
85. Mitchell H.B. (2007), *Multi-Sensor Data Fusion, An Introduction*, Springer Berlin Heidelberg, New York.
86. Milisavljevic Nada (2009), *Sensor and Data Fusion*, I-Tech Education and Publishing, Croatia.
87. Marzullo K. (1990), “Tolerating failures of continuous-valued sensors”, *ACM Transactions on Computer Systems* Vol. 8(4), pp. 284-304.
88. Murphy Robin R. (2000), *Introduction to AI Robotics*, A Brandford Book The MIT Press Cambridge, Massachusetts London, England.
89. Matthies L., Elfes A. (1988), “Integration of sonar and stereo range data using a grid-based representation”, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 727-733.
90. Martin M. C., Moravec H. P. (1996), *Robot evidence grids*, Technical Report CMU-RITR-96-06, The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA.
91. Nunez P., Drews J. P., Rocha R. , Dias J. (2009), “Data Fusion Calibration for a 3D Laser Range Finder and a Camera using Inertial Data”, *Proceedings of the 4th European Conference on Mobile Robots, ECMR'09*, pp. 31-36.
92. Ng G. W. (2004), *Intelligent Systems-Fusion, Tracking and Control*, UMIST, Research Studies Press Ltd, England.
93. Ng G. W., Ng K. H., (2000), “Sensor management - what, why and how”, *International Journal on Multi-Sensor, Multi-Source Information Fusion* Vol. 1(2), pp. 67-75.

94. Nelson V. P. (1990), "Fault-tolerant computing: Fundamental concepts", *IEEE Computer* Vol. 23(7), pp. 19-25.
95. Ohya A., Kosaka A., Kak A. (1998), "Vision-Based Navigation of Mobile Robot with Obstacle Avoidance by Single Camera Vision and Ultrasonic Sensing", *Proceedings of IEEE/RSJ International Conference*, pp. 704 -711.
96. Perrollaz M., Labayrade R., Royere C., Hautiere N., Aubert D. (2006), "Long Range Obstacle Detection Using Laser Scanner and Stereovision", *IEEE International Conference on Intelligent Vehicles Symposium*, pp. 182 -187.
97. Perceptron (2003), *ScanWorks 3D product Brochure*, www.perceptron.com.
98. Panich S., Afzulpurkar N. (2011), "Absolute Positioning Instruments for Odometry System Integrated with Gyroscope by Using IKF", *International Journal of Advanced Robotic Systems* Vol. 8(2), pp. 122-136.
99. Punska O. (1999), *Bayesian Approaches to Multi-Sensor Data Fusion*, the degree of Master of Philosophy, Department of Engineering University of Cambridge.
100. P. M. Duong, T. T. Hoang, N. T. T. Van, D. A. Viet and T. Q. Vinh (2012), "A Novel Platform for Internet-based Mobile Robot Systems", *The 7th IEEE Conference on Industrial Electronics and Applications*, pp.1969-1974.
101. Pfister S. T., Roumeliotis S. I., Burdick J. W. (2003), "Weighted line fitting algorithms for mobile robot map building and efficient data representation", *IEEE Conf. Robotics and Automation, Proceedings, ICRA '03*, pp. 1304-1311.
102. Premebida C., Nunes U. (2005), *Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications*, Institute of Systems and Robotics - ISR, Technical Report, Coimbra, Portugal.
103. Roberts J. M., Duff E. S., Corke P. I., Sikka P., Winstanley G. J., and Cunningham J. (2000), "Autonomous control of underground mining vehicles using reactive navigation", *Proceedings of the 2000 IEEE International Conference on Robotics & Automation*, pp. 3790-3795.

104. Ribeiro M. I., Goncalves J. G. M. (1996), "Natural Landmark based Localisation of Mobile Robots Using Laser range Data", *IEEE proceedings of EUROBOT'96*, pp. 186-193.
105. Secchi H., Carelli R., Mut V. (2003), "An Experience on Stable Control of Mobile Robots", *Latin American Applied Research* Vol. 33(4), pp. 379-385.
106. Selkainaho J. (2002), *Adaptive Autonomus Navigation Mobile Robots in Unknouwn Environments*, PhD Thesis of Science in Technology, Helsinki University of Technology, Finland.
107. Soumare Seydou, Ohya Akihisa and Yuta Shin'ichi (2002), "Real-Time Obstacle Avoidance by an Autonomous Mobile Robot using an Active Vision Sensor and a Vertically Emitted Laser Slit", *Intelligent Autonomuous Systems 7*, pp. 301- 308.
108. Surmann H., Lingemann K., Nuchter A., Hertzberg J. (2001), "A 3D laser range finder for autonomous mobile robots", *Proc. 32nd International Symposium on Robotics (ISR)*, pp. 153-158.
109. Steinberg A. N., Bowman C. L., White F. E. (1999), "Revisions to the JDL Data Fusion Model", *Sensor Fusion: Architectures, Algorithms, and Applications III* Vol. 3719(1), pp. 430-441.
110. Smestad T. (2001), *Data fusion - for humans, computers or both?*, Translated article from Mikroskopet, Norwegian Defence Research Establishment, Norway.
111. Siegwart R., Nourbkhsh I. R. (2004), *Introduction to Autonomous Mobile Robots*, A Brandford Book The MIT Press Cambridge, Massachusetts London, England.
112. Shaffer, G., Gonzalez, J., Stentz, A. (1992), "Comparison of two range-based pose estimators for a mobile robot," *Proceedings of SPIE Symposium on Mobile Robots*, pp. 661-667.

113. Sick AG. : 2006-08-01 Telegrams for Operating/ Configuring the LMS 2xx (Firmware Version V2.30/X1.27), www.sick.com, Germany.
114. Thrun S., Bucken A., Burgard W., Fox D., Frohlingshaus T., Hennig D., Hofman T., Krell M., and Schmidt Timo (1998), "Map learning and high-speed navigation in RHINO", *AI based Mobile Robots, MIT Press*, pp. 100-124.
115. Thrun S., Burgard W., Fox D. (2000), "Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping", *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 321-328.
116. Tran Thuan Hoang, Phung Manh Duong, Nguyen Van Tinh and Tran Quang Vinh (2012), "A Path Following Algorithm for Wheeled Mobile Robot Using Extended Kalman Filter", *IEICE Proc. of the 3th Int. Conf. on Integrated Circuit Design*, pp. 179-183.
117. Thomas Ciza (2011), Sensor Fusion - Foundation and Applications, *InTech Janeza Trdine 9*, Croatia.
118. Taylor C. (1991), "Building Representations for the Environment of a Mobile Robot from Image Data", *Proceedings of the SPIE Conference on Mobile Robots*, pp. 331-339.
119. Talluri R. and Aggarwal J. (1993), "Position Estimation Techniques for an Autonomous Mobile Robot- a Review". *In Handbook of Pattern Recognition and Computer Vision*, pp. 769-801.
120. Thomas Bayes T., Richard P. (1763), An Essay Towards Solving a Problem in Doctrine of Chances, *Philosophical Transactions* Vol. 53(1), pp. 370 - 418.
121. Tarin C., Brugger H., Moscardo R., Tibken B., Hofer E. P. (1999), "Low level sensor fusion for autonomous mobile robot navigation", *Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference*, pp. 1377-1382.

122. Teslic L., Skrjanc I., Klancar G. (2011), "EKF-Based Localization of a Wheeled Mobile Robot in Structured Environments", *Journal of Intelligent and Robotic Systems* Vol. 62(2), pp. 187-203.
123. Visser A., Groen F. C. A. (1999), *Organisation and design of autonomous systems*, Textbook, Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, Netherlands.
124. Wen Dai, Cuhadar Aysegul, and Liu Peter X. (2008), "Robot Tracking Using Vision and Laser Sensors", *IEEE 4th Conference on Automation Science and Engineering*, pp. 169 -174.
125. Wulf O., Wagner B. (2003), "Fast 3d scanning methods for laser measurement systems", *Proceedings of the International Conference on Control Systems and Computer Science*, pp. 312-317.
126. Widyotriatmo A., Hong Keum-Shik, and Prayudhi Ladin H. (2010), "Robust stabilization of a wheeled vehicle: Hybrid feedback control design and experimental validation", *Journal of Mechanical Science and Technology* Vol. 24(2), pp. 513-520.
127. Wald L. (1998), "A european proposal for terms of reference in data fusion", *International Archives of Photogrammetry and Remote Sensing* Vol. 32(7), pp. 651-654.
128. Waltz E. and Llinas J. (1990), *Multisensor Data Fusion*, Artech House, Norwood, Massachusetts.
129. Welch G. and Bishop G. (2006), *An Introduction to the Kalman Filter*, Department of Computer Science University of North Carolina, UNC-Chapel Hill, USA.
130. Yang H., Borenstein J., Wehe D. (2000), "Sonar-based Obstacle Avoidance for a Large, Non-point, Omni-directional Mobile Robot", *Proceedings of International Conference on Nuclear and Hazardous Waste Management*, pp. 21-28.

131. Yenilmez L., Temeltas H. (1998), “Real time multi-sensor fusion and navigation for mobile robots”, *9th Mediterranean Electrotechnical Conference*, pp. 221-225.
132. Young M. (1989), *The Technical Writer's Handbook*, University Science Books, Mill Valley, California.
133. Zhou Xiaowei, Ho Yeong Khing, Chua Chin Seng, Zou Yi (2000), “The Localization of Mobile Robot Based on Laser Scanner”, *Electrical and Computer Engineering*, pp. 841-845.
134. Zhang J., Wu Y., Liu W., Chen X. (2010), “Novel Approach to Position and Orientation Estimation in Vision-Based UAV Navigation”, *IEEE Transactions on Aerospace and electronic systems* Vol. 46(2), pp. 688-700.