

**ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN**

NÔNG MINH NGỌC

**NGHIÊN CỨU PHÁT TRIỂN KỸ THUẬT TRÁNH VA CHẠM
CHO ROBOT TỰ HÀNH**

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

Hà Nội – 2023

**ĐẠI HỌC QUỐC GIA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN**

NÔNG MINH NGỌC

**NGHIÊN CỨU PHÁT TRIỂN KỸ THUẬT TRÁNH VA CHẠM
CHO ROBOT TỰ HÀNH**

Chuyên ngành: Quản lí Hệ thống Thông tin

Mã số: 9480205.01QTD

LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC

- 1. PGS.TS. Đỗ Năng Toàn**
- 2. PGS.TS. Vũ Việt Vũ**

Hà Nội – 2023

LỜI CAM ĐOAN

Tôi xin cam đoan rằng nội dung luận án là kết quả nghiên cứu của riêng tác giả, được hoàn thành dưới sự hướng dẫn của PGS.TS Đỗ Năng Toàn và PGS.TS Vũ Việt Vũ. Tất cả những tham khảo từ các nghiên cứu liên quan đều được nêu nguồn gốc một cách rõ ràng, những kết quả nghiên cứu và đóng góp trong luận án chưa được công bố trong bất kì công trình khoa học nào khác.

Hà Nội, ngày 28 tháng 08 năm 2023

Tác giả luận án

Nghiên cứu sinh

Nông Minh Ngọc

LỜI CẢM ƠN

Trước hết, tác giả xin bày tỏ lòng biết ơn chân thành và sâu sắc tới tập thể cán bộ hướng dẫn, các thầy PGS.TS Đỗ Năng Toàn và PGS.TS Vũ Việt Vũ đã tận tình giúp đỡ, chỉ bảo và động viên trong suốt thời gian dài tác giả học tập, nghiên cứu và hoàn thiện luận án này.

Xin gửi lời cảm ơn chân thành đến các thầy giáo PGS.TS Ngô Như Khoa, TS. Vũ Vinh Quang, PGS.TS Phạm Thành Long, những người thầy đã chia sẻ, động viên tác giả thực hiện nội dung nghiên cứu và hoàn thiện luận án.

Xin chân thành cảm ơn tập thể lãnh đạo Viện Công nghệ thông tin- Đại học Quốc gia Hà Nội, tập thể lãnh đạo Đại học Thái Nguyên đã tạo điều kiện giúp đỡ tác giả thực hiện công việc học tập, nghiên cứu của mình.

Đặc biệt, xin gửi lời cảm ơn sâu sắc nhất tới gia đình, bạn bè và người thân, những người luôn động viên, chia sẻ và tạo điều kiện tốt nhất cho tác giả có thể học tập, nghiên cứu và hoàn thiện luận án này.

Hà Nội, ngày 28 tháng 08 năm 2023

Tác giả luận án

Nghiên cứu sinh

Nông Minh Ngọc

MỤC LỤC

LỜI CAM ĐOAN.....	i
LỜI CẢM ƠN.....	ii
MỤC LỤC.....	iii
DANH MỤC CHỮ VIẾT TẮT.....	vi
DANH MỤC BẢNG BIỂU.....	viii
DANH MỤC HÌNH ẢNH.....	ix
PHẦN MỞ ĐẦU	1
CHƯƠNG 1. TỔNG QUAN VỀ TRÁNH VÀ CHẠM CHO ROBOT TỰ HÀNH.....	6
1.1. Robot tự hành và những thách thức	6
1.1.1. Khái niệm về robot tự hành	9
1.1.2. Cấu tạo, nguyên lí hoạt động.....	10
1.1.3. Phân loại robot tự hành	11
1.1.3.1. Robot di chuyển bằng chân.....	11
1.1.3.2. Robot di chuyển bằng bánh xe.....	13
1.1.3.3. Robot khác	14
1.1.4. Các ứng dụng và thách thức của robot tự hành.....	15
1.1.4.1. Ứng dụng của robot tự hành	16
1.1.4.2. Thách thức đặt ra với robot tự hành.....	18
1.2. Một số tiếp cận về tránh và chạm	20
1.2.1. Tránh và chạm cho robot tự hành dựa trên lập kế hoạch.....	21
1.2.2. Tránh và chạm cho robot tự hành dựa trên phản ứng	23
1.3. Kết luận và vấn đề nghiên cứu.....	28
CHƯƠNG 2. KỸ THUẬT TRÁNH VÀ CHẠM DỰA TRÊN CẤU TRÚC CÂY PHÂN LỚP HỆ BAO BVH, KẾT HỢP THUẬT TOÁN ELASTIC STRIPS ..	29
2.1. Phương pháp phát hiện va chạm sử dụng hộp bao và cấu trúc hệ bao	29
2.1.1. Phát hiện, tính toán va chạm sử dụng hộp bao dạng trục	29

2.1.2. Phát hiện, tính toán va chạm sử dụng hộp bao theo hướng.....	30
2.1.2.1. Kiểm tra va chạm.....	32
2.1.2.2. Tìm điểm va chạm	37
2.1.3. Phát hiện, tính toán va chạm sử dụng hệ bao	43
2.2. Phương pháp tránh va chạm Elastic strips	46
2.2.1. Giới thiệu	46
2.2.2. Vấn đề tồn tại, hạn chế.....	58
2.3. Kỹ thuật phát hiện tránh va chạm dựa trên cấu trúc cây phân lớp hệ bao BVH, kết hợp với thuật toán Elastic strips.....	59
2.3.1. Đề xuất của phương pháp	59
2.3.2. Kết quả tính toán và mô phỏng	64
2.3.3. Nhận xét.....	66
2.4. Kết luận và vấn đề nghiên cứu.....	66
CHƯƠNG 3. KỸ THUẬT TRÁNH VA CHẠM DỰA TRÊN TÍNH TOÁN VÀ PHÂN VÙNG ĐỒNG MỨC XÁC SUẤT VA CHẠM.....	67
3.1. Xác suất va chạm và ứng dụng	67
3.1.1. Khái niệm	67
3.1.2. Ứng dụng	68
3.1.2.1. Phân tích độ nhạy của xác suất va chạm	68
3.1.2.2. Xác suất va chạm tối đa	69
3.1.2.3. Phân tích Thiếu và Báo động Sai.....	70
3.1.2.4. Đánh giá toàn diện rủi ro va chạm.....	71
3.2. Kỹ thuật phát hiện, tránh va chạm cho robot dựa trên tính toán xác suất.....	71
3.2.1. Cơ sở toán học.....	74
3.2.1.1. Các định nghĩa	74
3.2.1.2. Công thức tính toán va chạm trong không gian 2 chiều.....	76
3.2.1.3. Công thức tính toán va chạm trong không gian 3 chiều	78
3.2.2. Miền đồng mức xác suất va chạm.....	79
3.2.2.1. Các định nghĩa	79

3.2.2.2. Thuật toán xác định các miền đồng mức	80
3.2.3. Thuật toán tránh va chạm dựa trên xác suất va chạm.....	83
3.3. Kết quả tính toán và mô phỏng.....	84
3.3.1. Kết quả xác suất tránh va chạm P	84
a) Trường hợp 1	85
b) Trường hợp 2	86
c) Trường hợp 3	87
d) Trường hợp 4	88
3.3.2. Kết quả xác suất tránh va chạm (1-P)	89
a) Trường hợp 1	89
b) Trường hợp 2	90
3.3.3. Mô hình (1-P) với nhiều vật cản	91
3.2.4. Nhận xét.....	91
3.4. Kết luận và vấn đề nghiên cứu.....	92
PHẦN KẾT LUẬN.....	93
DANH MỤC CÔNG BỐ.....	95
TÀI LIỆU THAM KHẢO	96

DANH MỤC CHỮ VIẾT TẮT

Từ viết tắt	Từ viết đầy đủ	Ghi chú
AABB	Axis-Aligned Bounding Boxe	
AGV	Automated Guided Vehicle	
AI	Artificial Intelligence	
AMR	Autonomous Mobile Robot	
APF	Artificial Potential Field	
AS/RS	Unitload AS/RS	
BVH	Bounding Volume Hierarchies	
CAGR	Compound Annual Growth Rate	
CPS	Cyber-Physical Systems	
CRI	Collision Risk Index	
CV	Cell value	
DCPA	Distance of the Closest Point of Approach	
GPF	Generalized Potential Field	
GPF	Generalized Potential Fields	
GPS	Global Positioning System	
GTP	Goods to People	
HRI	Human Robot Interaction	
IFR	International Federation of Robotics	
IoT	Internet of Things	
kDOPs	k-Discrete Oriented Polytopes	
LHC	Large Hadron Collider	
LIDAR	Light Detection And Ranging	
LIDAR	LASER RANGEFINDER	
MDP	Markov Decision Process	
MPC	Model Predictive Control	

OBB	Oriented Bounding Boxe	
PVO	Probabilistic Velocity Obstacles	
SLAM	Simultaneous Localization And Mapping	
SMDP	Semi-Markov Decision Process	
STEM	Science-Technology-Engineering-Maths	
TCA	Time of Closest Approach	
TCPA	Time to the Closest Point of Approach	
UAV	Unmanned Aerial Vehicle	
UKF	Unscented Kalman Flter	
VFF	Virtual Force Field	
VO	Velocity Obstacles	

DANH MỤC BẢNG BIỂU

Bảng 2.1. Bảng các giá trị R, R0, R1 được tính toán trước.	36
Bảng 2.2. Bảng tính toán tọa độ của tiếp điểm trong mọi trường hợp.....	41
Bảng 2.3. Bảng dữ liệu mô phỏng cho robot	64
Bảng 3.1. Bảng tham số tính toán xác suất va chạm.....	84
Bảng 3.2. Dữ liệu tính toán xác suất phân bù va chạm (1-P).....	89

DANH MỤC HÌNH ẢNH

Hình 1.1. Thị trường công nghệ robot, 2022-2030 (đơn vị: tỷ USD)[73]	6
Hình 1.2. Thị trường công nghệ robot theo khu vực (đơn vị: %) [73]	7
Hình 1.3. Ứng dụng của công nghệ robot	8
Hình 1.4. Robot tự hành di chuyển bằng chân [31]	12
Hình 1.5. Robot tự hành di chuyển bằng bánh xe.....	13
Hình 1.6. Ứng dụng của robot tự hành [2].....	17
Hình 1.7. Những thách thức của robot tự hành [2]	18
Hình 1.8. Vấn đề điều hướng của robot tự hành	19
Hình 1.9. Phương pháp chia lưới xác định [8].....	24
Hình 1.10. Phương pháp trường thế năng nhân tạo	25
Hình 1.11. Phương pháp trường lực ảo (VFF) [10]	27
Hình 2.1. Hộp bao AABB của đối tượng [24]	29
Hình 2.2. Hộp bao OBB của đối tượng [24]	31
Hình 2.3. Hình chiếu của P lên đường thẳng d	32
Hình 2.4. Chiều 8 đỉnh của hình hộp lên trục cô lập d.....	33
Hình 2.5. Kết quả chiếu 2 hình hộp lên trục cô lập d [24].....	35
Hình 2.6. Tìm điểm va chạm khi hai đối tượng tiếp xúc nhau [24].....	37
Hình 2.7. Hệ bao BVH sử dụng phương pháp khối bao chữ nhật.	44
Hình 2.8. Xây dựng hệ bao BVH sử dụng phương pháp Median-Cut.....	44
Hình 2.9. Định nghĩa trường lực tiềm năng	47
Hình 2.10. Mô hình robot và hệ trục tọa độ.....	48
Hình 2.11. Cấu trúc của Elastic strips	49
Hình 2.12. Nội lực.....	50
Hình 2.13. Sự thay đổi quỹ đạo chuyển động của robot [67]	51
Hình 2.14. Ngoại lực	52
Hình 2.15. Không gian an toàn cho robot	52
Hình 2.16. Tính toán vị trí vật cản	53

Hình 2.17. Nội lực và Ngoại lực của Elastic strips	55
Hình 2.18. Tổng hợp lực Elastic strips.....	55
Hình 2.19. Sơ đồ thuật toán tránh va chạm dựa trên Elastic strips	57
Hình 2.20. Quỹ đạo chuyển động của robot với phương pháp Elastic strips	58
Hình 2.21. Mô tả tổng hợp lực Elastic strips gốc khi hướng di chuyển hiện tại trùng với hướng điểm va chạm.....	59
Hình 2.22. Hộp bao của robot OBB-R.....	60
Hình 2.23. Hộp bao của vật cản OBB-O.....	61
Hình 2.24. Xác định thành phần ngoại lực của phương pháp cải tiến	63
Hình 2.25. Sơ đồ thuật toán cải tiến Elastic strips	64
Hình 2.26. Kết quả mô phỏng trong trường hợp 1	65
Hình 2.27. Kết quả mô phỏng trong trường hợp 2,3,4 với hai vật cản	65
Hình 2.28. Kết quả mô phỏng trong trường hợp 2,3,4 với ba vật cản	66
Hình 3.1. Vùng va chạm	76
Hình 3.2. Cơ sở hình học tính toán va chạm 2D.....	76
Hình 3.3. Sơ đồ thuật toán tránh va chạm dựa trên xác suất va chạm.....	83
Hình 3.4. Kết quả trường hợp 1	85
Hình 3.5. Kết quả trường hợp 2	86
Hình 3.6. Kết quả trường hợp 3	87
Hình 3.7. Kết quả trường hợp 4	88
Hình 3.8. Kết quả trường hợp 1 với 1-P	89
Hình 3.9. Kết quả trường hợp 2 với 1-P	90
Hình 3.10. Kết quả tính toán và mô phỏng (1-P) cho nhiều vật cản.....	91

PHẦN MỞ ĐẦU

1. Lí do chọn đề tài

Công nghệ robot (Robotics) đang là tâm điểm của cuộc cách mạng Công nghiệp lần thứ Tư và là sản phẩm công nghệ có độ phức tạp cao, chứa hàm lượng tri thức vô cùng phong phú về tất cả các lĩnh vực của khoa học và công nghệ như công nghiệp, nông nghiệp, y tế, giáo dục đào tạo, giải trí, an ninh quốc phòng, thám hiểm không gian. Những năm gần đây, sự phát triển như vũ bão của khoa học kỹ thuật đã tác động sâu sắc đến các ngành nghề nói chung và ngành công nghiệp nói riêng. Đã có không ít các doanh nghiệp thực hiện việc triển khai và ứng dụng kho hàng thông minh nhằm mục tiêu tiết kiệm chi phí, giảm thiểu lao động thủ công, tăng cường tự động hóa, tối đa hóa lợi nhuận.

Theo tổng kết của tác giả McFarlane¹, thuật ngữ **“kho hàng thông minh”** đề cập đến các hoạt động quản lý kho hàng bao gồm việc lên kế hoạch, quản lý hoặc kiểm soát kho hàng theo cách thông minh hơn, hiện đại hơn so với các giải pháp thông thường. Cùng với đó, tác giả Wen, Jinming² đã đưa ra nhận định về nhu cầu ứng dụng công nghệ tự động hóa và công nghệ thông minh trong quản lý kho hàng. Và từ đó, khái niệm kho hàng thông minh được hiểu là khu vực lưu trữ nguyên vật liệu và hàng hóa, trong đó có sử dụng kết hợp nhiều công nghệ tự động, được kết nối với nhau tạo ra một môi trường công nghệ, sự kết hợp giữa các công nghệ không chỉ trong phần kho, mà còn trong hoạt động quản lý và vận hành kho nhằm tăng năng suất, chất lượng và hiệu quả sản xuất.

Mục tiêu của kho hàng thông minh là cung cấp cho các doanh nghiệp khả năng hiển thị theo thời gian thực, tối ưu hóa các quy trình và thực hiện các thay đổi tức thời, cải thiện tốc độ hoạt động, loại bỏ phần lớn lỗi của con người và tăng cường an toàn và bảo mật trong sản xuất. Chính bởi các mục tiêu này, kho hàng thông minh

¹ “*Intelligent logistics: Involving the customer*”

² “*Swarm robotics control and communications: Imminent challenges for next generation smart logistics*”

giúp doanh nghiệp tiết kiệm thời gian, tài chính, tăng lợi nhuận, nâng cao mức độ hài lòng của khách hàng và tăng giá trị thương hiệu của doanh nghiệp. Sự ra đời của hàng loạt hệ thống tự động với mục đích thay thế sức lao động của con người, tăng năng suất lao động và tự động hóa trong kho hàng thông minh cũng là một sản phẩm không nằm ngoài mục đích đó. Trong nhiều thập kỉ qua, quản lí kho đã phát triển thành một khoa học phức tạp và công nghệ tự động hóa kho giúp cải thiện hiệu suất, xúc tác trong hoạt động của chuỗi cung ứng. Do vậy, nhiều mô hình tự động hóa ra đời để khắc phục những rủi ro của mô hình truyền thống, mang lại hiệu quả rất lớn trong công tác quản lí kho.

Tự động hóa kho hàng là quá trình áp dụng các hệ thống công nghệ (điều khiển thông minh, máy móc, thiết bị công nghệ cao và robot) trong vận hành, quản lí kho nhằm cắt bỏ các công việc thủ công, giảm sai sót, nhầm lẫn, tăng hiệu suất trong khâu vận hành (vận chuyển, đóng gói, kiểm soát và lưu trữ) và tăng tính chính xác cho việc quản lí kho hàng. Mức độ tự động hóa kho hàng được phân loại theo độ tự động của hệ thống quản lí kho như: tự động hóa cơ bản, tự động hóa bằng cơ giới hóa, tự động hóa nâng cao, tự động hóa hoàn toàn.

Các doanh nghiệp hiện nay luôn hướng tới việc tối ưu hóa lợi nhuận, nâng cao hiệu quả kinh tế, đảm bảo chất lượng hàng hóa. Do vậy, xu hướng tự động hóa kho hàng là hết sức thiết yếu, đòi hỏi doanh nghiệp đầu tư, trang bị các hệ thống, thiết bị tự động để nâng cao quy trình sản xuất.

Tự động hóa kho hàng tạo nên mô hình quản lí kho tự động có sự tham gia của hệ thống các **robot tự hành, robot thông minh** nhằm tự động hóa các hoạt động vận hành trong kho hàng thông minh. Các robot này có khả năng hoạt động theo chương trình được lập trình sẵn với tốc độ và độ chính xác cao, có khả năng tiếp nhận tri thức và tư duy giống như con người để phán đoán và ra quyết định mà không cần đến sự can thiệp của con người. Việc triển khai các robot tự hành trong nhà kho mang lại nhiều lợi ích. Đó là, tăng hiệu quả, nâng cao năng suất, cải thiện độ chính xác, tính linh hoạt, khả năng mở rộng, cải thiện an toàn và truy cập vào dữ liệu và phân tích

thời gian thực. Những lợi thế này dẫn đến tiết kiệm chi phí, dịch vụ khách hàng tốt hơn và lợi thế cạnh tranh cho các nhà kho trong bối cảnh kinh doanh năng động.

Tuy nhiên, doanh nghiệp sẽ gặp phải một số bất cập khi triển khai hệ thống robot tự hành trong hoạt động quản lý và vận hành kho như: đòi hỏi đầu tư ban đầu đáng kể (chi phí mua robot, tích hợp hệ thống, đào tạo cán bộ vận hành,...); việc thiết lập và bảo trì các hệ thống phức tạp dựa trên các công nghệ tiên tiến như cảm biến, thuật toán điều hướng và mạng truyền thông đòi hỏi đội ngũ nhân viên phải có kỹ thuật chuyên môn cao; việc thích ứng với các tình huống không cho trước hoặc xử lý trường hợp ngoại lệ nằm ngoài khả năng được lập trình của robot cũng là một nhược điểm trong quá trình vận hành hệ thống kho.

Cùng với đó, robot tự hành hiện nay có sự phát triển vượt bậc, được ứng dụng rộng rãi xong còn bao chứa nhiều thách thức nhằm hoàn thiện và đáp ứng tối đa yêu cầu của con người, của thực tế. Trong đó, thách thức không hề nhỏ phải là giải quyết vấn đề tránh va chạm cho robot tự hành trong quá trình hoạt động. Đây cũng đang thu hút sự quan tâm nghiên cứu không chỉ của nhiều tập đoàn công nghệ lớn như Boston Dynamics, ABB Robotics, KUKA Robotics, Yaskawa,... mà còn của nhiều nhà khoa học trên thế giới.

2. Mục đích nghiên cứu

Luận án nghiên cứu, phát triển các kỹ thuật tránh va chạm cho robot tự hành dựa trên các thông tin như quỹ đạo, vận tốc, kích thước của robot. Cụ thể:

- Nghiên cứu và phát triển thuật toán tránh va chạm cho robot dựa trên cấu trúc cây phân lớp hệ bao BVH kết hợp với kỹ thuật Elastic strips.

- Nghiên cứu và đề xuất thuật toán tránh va chạm cho robot dựa trên tính toán xác suất và phân vùng đồng mức xác suất va chạm cho robot tự hành.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: các phương pháp phát hiện, tránh va chạm cho robot.

Phạm vi nghiên cứu: phương pháp phát hiện, tránh va chạm cho robot tự hành; kỹ thuật Elastic strips; kỹ thuật phát hiện, tính toán va chạm hộp bao theo hướng; phương

pháp tính toán va chạm dựa trên xác suất va chạm và phân vùng đồng mức xác suất va chạm.

4. Nội dung nghiên cứu

Luận án tập trung nghiên cứu các nội dung:

- Tổng quan về robot tự hành, ứng dụng, thách thức và vấn đề tránh va chạm.
- Nghiên cứu và phát triển kĩ thuật tránh va chạm cho robot tự hành dựa trên cấu trúc cây phân lớp hệ bao (BVH), kết hợp với thuật toán Elastic strips nhằm nâng cao hiệu quả phát hiện, tránh va chạm.
- Nghiên cứu và đề xuất kĩ thuật phát hiện, tránh va chạm cho robot tự hành dựa trên mô hình tính toán xác suất va chạm, phân vùng đồng mức xác suất va chạm.
- Các mô hình tính toán xác suất va chạm dựa trên mô hình toán học thể tích; thuật toán phân vùng đồng mức va chạm, mô hình xác suất va chạm (P) và phân bù xác suất va chạm (1-P) của các đối tượng;

5. Phương pháp nghiên cứu

- Phương pháp tổng kê tổng hợp: Phương pháp này được sử dụng để thu thập, tổng hợp các tài liệu kĩ thuật, các công bố khoa học các tài liệu liên quan đến kho hàng thông minh, tự động hóa kho hàng, robot tự hành và các kĩ thuật phát hiện, tránh va chạm cho robot tự hành. Từ đó, luận án phân tích và lựa chọn hướng tiếp cận khoa học, phù hợp.

- Nghiên cứu lí thuyết: Phương pháp này được sử dụng để tìm hiểu về các tài liệu kĩ thuật, các công trình nghiên cứu, các bài báo khoa học về tìm đường, phát hiện, tránh va chạm cho robot tự hành, các kĩ thuật tính toán va chạm dựa trên xác suất va chạm. Từ đó, luận án tiến hành phân tích, tổng hợp và đưa ra các vấn đề cần nghiên cứu.

- Phương pháp chuyên gia: Phương pháp nghiên cứu chuyên gia được sử dụng bằng cách tham gia các hội thảo khoa học nhằm trao đổi các kinh nghiệm, thu thập các ý kiến đóng góp của các chuyên gia và tích cực trao đổi với các chuyên gia nước ngoài.

- Phương pháp mô hình hóa, mô phỏng: Phương pháp này sử dụng các mô hình toán học cho robot, mô phỏng và kiểm chứng các thuật toán xử lý số liệu, thuật toán mô phỏng trên Matlab, lập trình trên các ngôn ngữ C, C++.

- Phương pháp kiểm chứng: Phương pháp này được sử dụng kết hợp phân tích đánh giá thông qua các kết quả tính toán, kết quả mô phỏng.

6. Đóng góp của luận án

1- Đề xuất thuật toán phát hiện, tránh va chạm cho robot tự hành dựa trên cấu trúc cây phân lớp hệ bao BVH, kết hợp với thuật toán Elastic strips nhằm nâng cao hiệu quả của việc phát hiện và tránh va chạm;

2- Đề xuất mô hình tính toán xác suất va chạm dựa trên mô hình toán học thể tích; thuật toán phân vùng đồng mức va chạm, mô hình xác suất va chạm (P) và phần bù xác suất va chạm (1-P) của các đối tượng và thuật toán phát hiện, tránh va chạm cho robot tự hành dựa trên phân vùng đồng mức va chạm.

7. Bố cục luận án

Cấu trúc luận án gồm: Phần mở đầu, 3 Chương nội dung và Phần kết luận.

Phần Mở đầu: trình bày mục đích nghiên cứu của luận án, đối tượng và phạm vi nghiên cứu luận án; nội dung và phương pháp nghiên cứu của luận án; những đóng góp của luận án.

Chương 1: Tổng quan về tránh va chạm cho robot tự hành.

Chương 2: Kỹ thuật tránh va chạm dựa trên cấu trúc cây phân lớp hệ bao BVH kết hợp với thuật toán Elastic strips.

Chương 3: Kỹ thuật tránh va chạm dựa trên tính toán và phân vùng đồng mức xác suất va chạm.

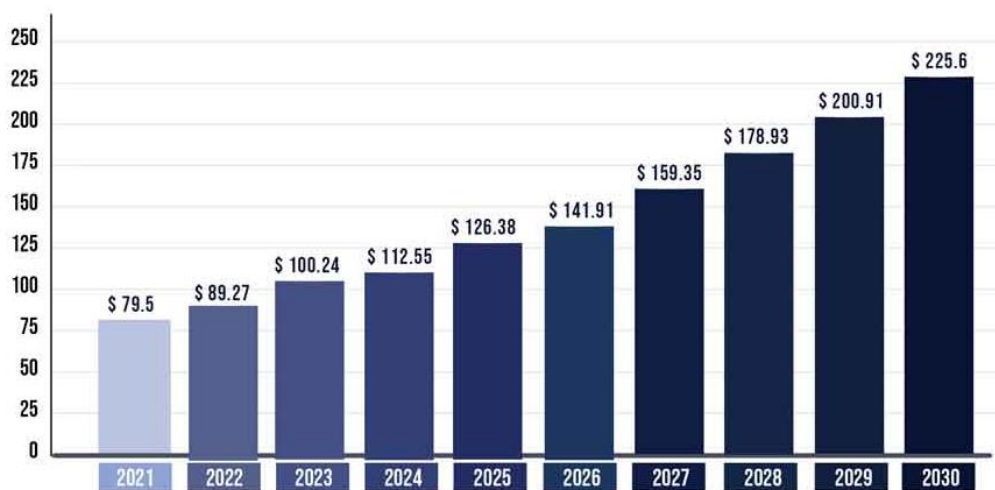
Phần Kết luận: nêu những đóng góp, hướng phát triển và những vấn đề quan tâm; danh mục các công trình đã được công bố của luận án; danh sách các tài liệu tham khảo được sử dụng trong luận án.

CHƯƠNG 1. TỔNG QUAN VỀ TRÁNH VA CHẠM CHO ROBOT TỰ HÀNH

1.1. Robot tự hành và những thách thức

Cuộc cách mạng công nghiệp lần thứ tư hay còn gọi là công nghiệp 4.0 với sự kết hợp của công nghệ như dữ liệu lớn, vạn vật kết nối, trí tuệ nhân tạo, tự động hóa quy trình robotic đã tác động sâu sắc đến đời sống kinh tế, chính trị, xã hội của thế giới. Công nghiệp 4.0 đã làm thay đổi nền tảng, thúc đẩy các nền kinh tế truyền thống chuyển đổi sang kinh tế tri thức và đổi mới sáng tạo, thúc đẩy sự phát triển không ngừng của khoa học công nghệ, khoa học kỹ thuật trong đó có công nghệ robot.

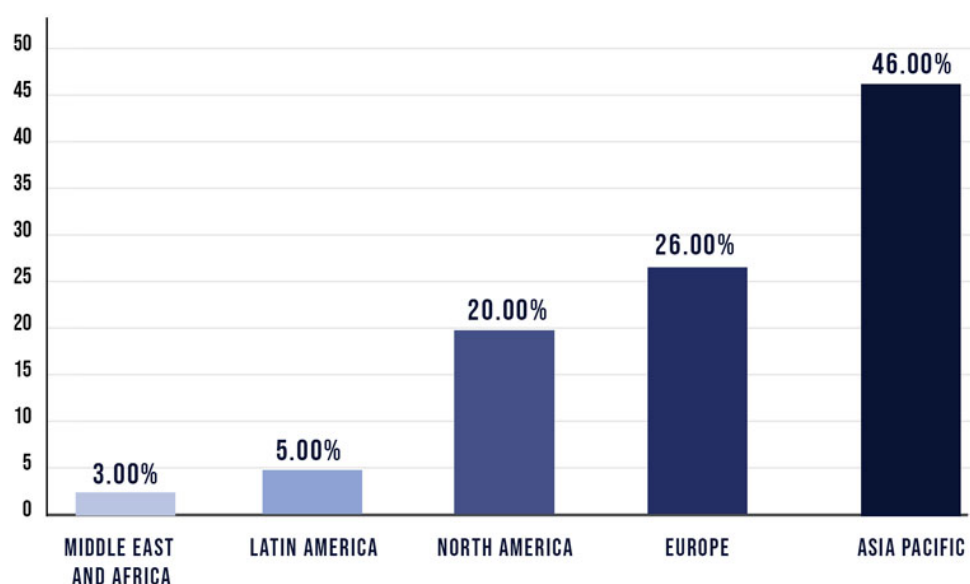
Công nghệ robot là một lĩnh vực khoa học, kỹ thuật chuyên nghiên cứu, thiết kế và phát triển các robot có khả năng thực hiện các nhiệm vụ và tác động vào môi trường xung quanh thông qua cảm biến, xử lý thông tin và hành động vật lý. Cùng với sự kết hợp bởi các lĩnh vực cơ điện tử, điện tử, công nghệ thông tin, trí tuệ nhân tạo và các hệ thống điều khiển thì sản phẩm của công nghệ robot ngày càng phát triển, có khả năng hoạt động thông minh, linh hoạt và tương tác với con người.



Hình 1.1. Thị trường công nghệ robot, 2022-2030 (đơn vị: tỷ USD)[73]

Theo báo cáo "World Robotics" của Liên đoàn robot quốc tế (IFR) [73], thị trường công nghệ robot toàn cầu năm 2021 ước tính khoảng 79,5 tỷ USD, năm 2022 là 89,27 tỷ USD và dự kiến đến năm 2030 sẽ vượt 225,6 tỷ USD với tỉ lệ tăng trưởng hàng năm (từ năm 2022 đến 2030) CAGR): 12,29% (Hình 1.1).

Kết quả đánh giá thị trường công nghệ robot theo khu vực năm 2021 (Hình 1.2) cho thấy, khu vực Châu Á- Thái Bình Dương dẫn đầu với các quốc gia như Trung Quốc, Hàn Quốc và Ấn Độ. Khu vực này đang gia tăng ứng dụng robot trong cả lĩnh vực công nghiệp và hộ gia đình. Khu vực Châu Âu đứng vị trí số 2, trong đó nước Anh với ưu thế chủ yếu là robot giáo dục, robot công nghiệp, robot giải trí tương tác và robot dịch vụ; Đan Mạch, Đức, Thụy Điển và Ý với các robot ứng dụng trong công nghiệp, nông nghiệp.



Hình 1.2. Thị trường công nghệ robot theo khu vực (đơn vị: %) [73]

Công nghệ robot ngày càng được ứng dụng đa dạng hơn, rộng rãi hơn trong các lĩnh vực của đời sống xã hội. Chúng mang lại lợi ích to lớn cho con người bằng việc gia tăng năng suất, chất lượng và an toàn trong quá trình sản xuất; chăm sóc sức khỏe (thực hiện phẫu thuật phức tạp, hỗ trợ người khuyết tật, người); hỗ trợ và chăm sóc khách hàng (tư vấn, vận chuyển, đóng gói hàng hóa); thám hiểm và thu thập dữ liệu bên ngoài không gian hoặc những nơi con người không đặt chân đến được; hỗ trợ đào tạo, giảng dạy và truyền cảm hứng cho học sinh trong lĩnh vực giáo dục; ứng dụng trong quốc phòng và an ninh để giám sát, xử lý và thực hiện các nhiệm vụ quốc phòng như xử lý bom, UAV, phương tiện tự hành,...(Hình 1.3).



Hình 1.3. Ứng dụng của công nghệ robot

Cùng với sự phát triển của công nghệ robot nói chung, robot tự hành cũng đang trở thành một đối tượng được quan tâm nghiên cứu, phát triển trong những năm gần đây bởi khả năng thay thế con người thực hiện các công việc mang tính lặp đi lặp lại, trong các môi trường độc hại, nguy hiểm và trong nhiều lĩnh vực. Theo Allied Market Research [72], năm 2018 thị trường robot tự hành đạt 9.340 triệu đô la và dự kiến sẽ đạt 39.585 triệu đô la vào năm 2026, với tỷ suất tăng trưởng hàng năm từ năm 2019 đến năm 2026 (CAGR) là 21,5%. Trong đó, thị trường Bắc Mỹ lớn nhất toàn cầu với 3.933,1 triệu đô la vào năm 2018 và dự kiến đạt 14.492,1 triệu đô la vào năm 2026.

Nghiên cứu, chế tạo robot tự hành là lĩnh vực của khoa học và công nghệ nhằm mục tiêu tạo ra các robot có thể di chuyển trong mọi môi trường mà không cần sự can thiệp của con người. Chúng có thể tự động di chuyển và làm việc (trong nhà máy công nghiệp, phòng thí nghiệm, bề mặt hành tinh, vv thông qua các thiết bị như cảm biến, phần mềm và các thiết bị các được gắn kèm trên robot. Hiện nay, các vấn đề cơ

bản của robot tự hành đang được quan tâm, tập trung nghiên cứu là chuyển động, cảm biến và điều hướng.

1.1.1. Khái niệm về robot tự hành

Robot tự hành là một loại robot có khả năng di chuyển và thực hiện các nhiệm vụ một cách độc lập, mà không cần sự can thiệp, điều khiển của con người. Các robot tự hành được trang bị các công nghệ như máy tính, cảm biến, trí tuệ nhân tạo, học sâu, và lý thuyết điều khiển tự động để có thể tự động hoá các tác vụ như di chuyển, phát hiện và nhận biết môi trường, thu thập thông tin, định vị, xử lý dữ liệu, và ra quyết định..).

Các ứng dụng của robot tự hành rất đa dạng, từ tự động hoá sản xuất, dịch vụ tư vấn, vận chuyển, dọn dẹp, y tế, nông nghiệp, đến khám phá không gian và các môi trường nguy hiểm. Robot tự hành đang được phát triển nhanh chóng và được xem là một trong những công nghệ tiên tiến nhất hiện nay. Chúng ngày càng hoàn thiện và được sử dụng thay thế con người trong các môi trường làm việc khác nhau.

Hiện nay, nghiên cứu, chế tạo robot tự hành có 2 hướng: hướng thứ nhất là nghiên cứu về robot tự hành có khả năng điều hướng ở tốc độ cao, có khả năng hoạt động ở đa môi trường (trong nhà và ngoài trời). Loại robot này yêu cầu khả năng tính toán đồ sộ và được trang bị cảm biến có độ nhạy cao, dải đo lớn để có thể điều khiển robot di chuyển ở tốc độ cao, trong môi trường có địa hình phức tạp. Hướng thứ hai là giải quyết các vấn đề của robot hoạt động trong môi trường làm việc, trong không gian cố định, giới hạn. Loại robot tự hành này có kết cấu đơn giản hơn loại trên, thực hiện những nhiệm vụ đơn giản.

Cùng với những hướng nghiên cứu chủ yếu ở trên, việc giải quyết một vấn đề quan trọng với robot tự hành là bài toán dẫn hướng cho robot tự hành cũng được chia làm 2 nhóm tương tự là dẫn hướng toàn cục (global) với môi trường làm việc của robot hoàn toàn xác định, đường đi và vật cản là hoàn toàn biết trước và dẫn hướng cục bộ (local) ứng với môi trường hoạt động của robot không biết trước hoặc chỉ biết một phần. Việc dẫn hướng cục bộ cần sự hỗ trợ thông tin phản ánh môi trường từ các cảm biến và thiết bị định vị giúp robot xác định được vật cản, vị trí và điểm mục tiêu.

1.1.2. Cấu tạo và nguyên lý hoạt động

Robot tự hành có cấu trúc vật lý hoặc khung chứa các thành phần khác nhau. Khung gồm được trang bị các cơ cấu di động như bánh xe, đường ray hoặc chân, tùy thuộc vào thiết kế và ứng dụng dự định của robot.

Bộ phận khung, vỏ: có tác dụng liên kết các bộ phận khác của robot. Đồng thời, nó cũng có tác dụng trong việc tăng độ cứng vững, bảo vệ robot khỏi các tác động từ bên ngoài.

Bộ phận cảm biến: bao gồm các cảm biến giúp nhận biết môi trường hoạt động và hỗ trợ quá trình điều khiển robot.. Những cảm biến này có thể là hình ảnh, LiDAR, cảm biến siêu âm, cảm biến hồng ngoại hoặc các loại cảm biến tiệm cận khác với nhiệm vụ thu thập dữ liệu về môi trường xung quanh robot (bao gồm chướng ngại vật, địa hình hoặc đối tượng quan tâm) để từ đó cung cấp dữ liệu, thông tin đầu vào cho robot xử lý, đưa ra các quyết định, và thực hiện điều khiển các cơ quan chấp hành thực thi theo kết quả tính toán (bao gồm: tránh vật cản, điều chỉnh tốc độ, thực thi các nhiệm vụ khác).

Bộ phận mạch điều khiển: có vai trò điều khiển hoạt động của mobile robot thông qua quá trình xử lý tín hiệu từ cảm biến, điều khiển động cơ và giao tiếp với máy tính (nếu có). Bộ phận này bao gồm các thành phần: phần cứng là bộ vi điều khiển hoặc bộ vi xử lý, bộ điều khiển động cơ và các thành phần điện tử khác. phần mềm là các thuật toán và chương trình điều khiển chuyển động và ra quyết định của robot. Khi tiếp nhận được dữ liệu từ hệ thống cảm biến, hệ thống điều khiển căn cứ vào nhiệm vụ và yêu cầu cụ thể, thực hiện các thuật toán để tính toán, xử lý như tính toán điểm va chạm, áp dụng thuật toán tránh va chạm cho robot, tính toán vị trí di chuyển, thực hiện chương trình điều khiển hệ thống động cơ, hệ thống chấp hành.

Bộ phận truyền động: robot tự hành được dẫn động bằng các bánh xe, dây đai, cánh, xích hay các khớp dịch chuyển. Sau khi tiếp nhận các thông tin và yêu cầu ra quyết định từ hệ thống điều khiển, robot tự hành thực hiện các tác vụ cụ thể về điều hướng, chuyển động liên quan đến việc phát hiện, tránh chướng ngại vật, tương tác giữa người và robot.

Bộ phận công suất: cung cấp năng lượng cho robot hoạt động. Bộ phận này phải cung cấp công suất đủ lớn cho toàn hệ thống robot hoạt động bao gồm công suất động cơ, công suất mạch điện cho hệ thống cảm biến, hệ thống điều khiển...

Các bộ phận này tạo thành thể thống nhất, hoạt động nhịp nhàng trình hoạt động của robot: 1- robot liên tục nhận phản hồi từ các cảm biến, định vị vị trí hiện tại, vị trí cần đến; 2- so sánh phản hồi đó với hành vi mong muốn; 3- thực hiện việc tính toán các giá trị về vị trí, về vận tốc, về góc quay thông qua các thuật toán điều khiển; 4- điều hướng, điều khiển các chuyển động, tránh chướng ngại vật, thực hiện các tác vụ và tương tác với môi trường xung quanh. Quá trình này được lặp lại cho đến khi robot hoàn thành nhiệm vụ được giao.

1.1.3. Phân loại robot tự hành

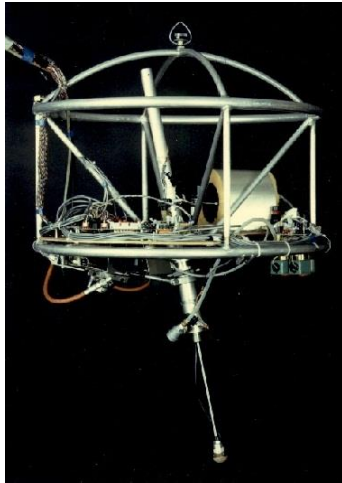
Robot tự hành được chia làm 2 loại chính đó là: robot tự hành di chuyển bằng chân và robot tự hành di chuyển bằng bánh xe [31]. Ngoài ra, một số loại robot hoạt động trong các môi trường đặc biệt (như dưới nước hay trên không trung) được trang bị cơ cấu di chuyển đặc trưng [73].

1.1.3.1. Robot di chuyển bằng chân

Robot tự hành di chuyển bằng chân được mô phỏng theo các loài động vật vì thế mà chúng có loại 1 chân, loại 2,4,6 chân và có thể nhiều hơn. Hình 1.4 giới thiệu một số loại robot điển hình chuyển động bằng chân.

Ưu điểm của loại robot này là có thể thích nghi và di chuyển trên các địa hình gồ ghề và có thể đi qua những vật cản như hố, vết nứt sâu.

Nhược điểm của loại robot là việc chế tạo quá phức tạp, chân robot là kết cấu nhiều bậc tự do và là nguyên nhân làm tăng trọng lượng của robot, giảm tốc độ di chuyển. Bởi vậy, robot loại này càng linh hoạt thì chi phí chế tạo càng cao.



a) Robot 1 chân Raibert



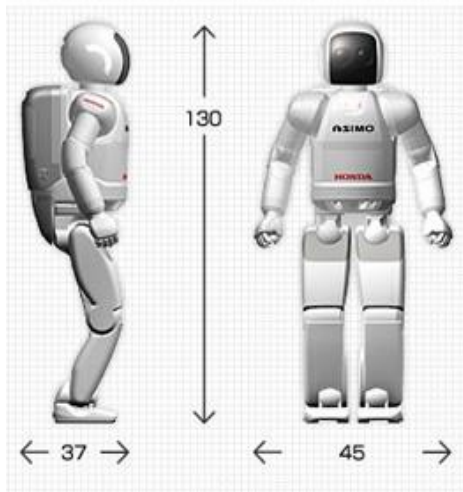
b) Robot 6 chân



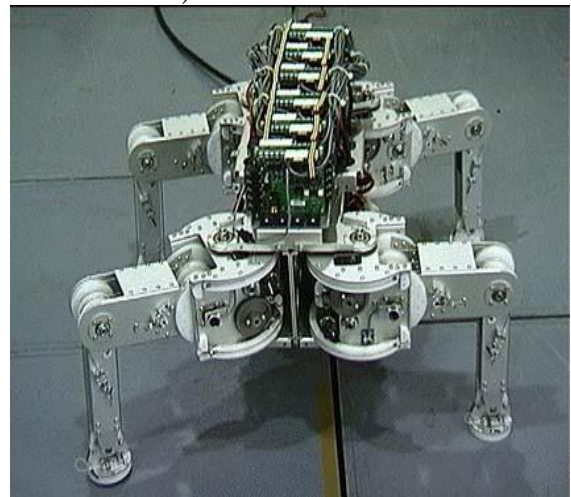
c) Robot SDR-4X



d) Robot chó Tian



e) Robot ASIMO

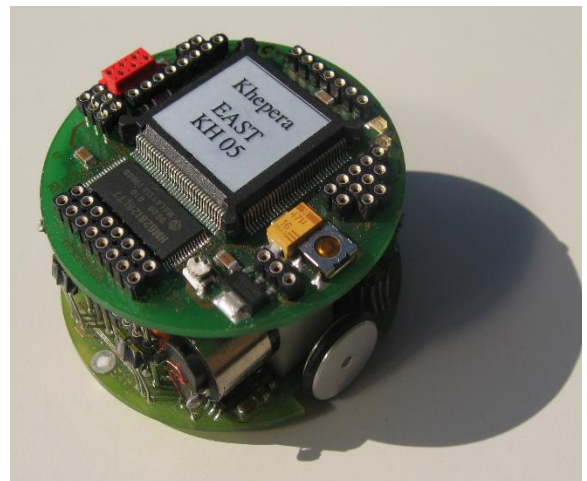
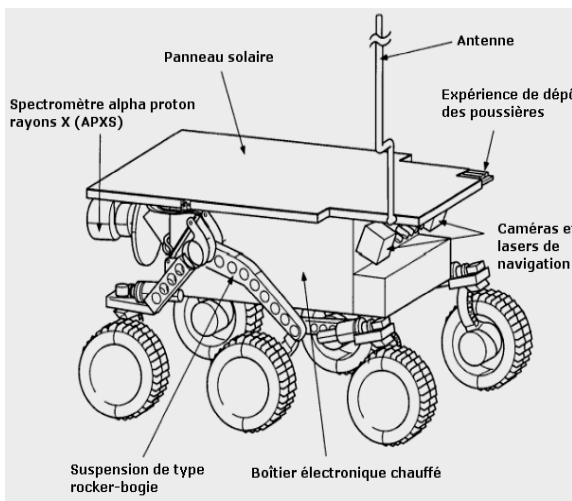


f) Robot 4 chân

Hình 1.4. Robot tự hành di chuyển bằng chân [31]

1.1.3.2. Robot di chuyển bằng bánh xe

Bánh xe là cơ cấu chuyển động được sử dụng rộng rãi nhất trong công nghệ chế tạo robot tự hành. Vấn đề cân bằng thường không phải là vấn đề được chú ý nhiều trong robot di chuyển bằng bánh. Ba bánh là kết cấu có khả năng duy trì cân bằng nhất, tuy nhiên kết cấu 2 bánh cũng có thể cân bằng được. Khi robot có số bánh nhiều hơn 3 thì thông thường người ta phải thiết kế hệ thống treo để duy trì sự tiếp xúc của tất cả các bánh xe với mặt đất. Đây là cách thức di chuyển phổ biến nhất, phù hợp với nhiều nền tảng và kích thước khác nhau của robot (Hình 1.5).

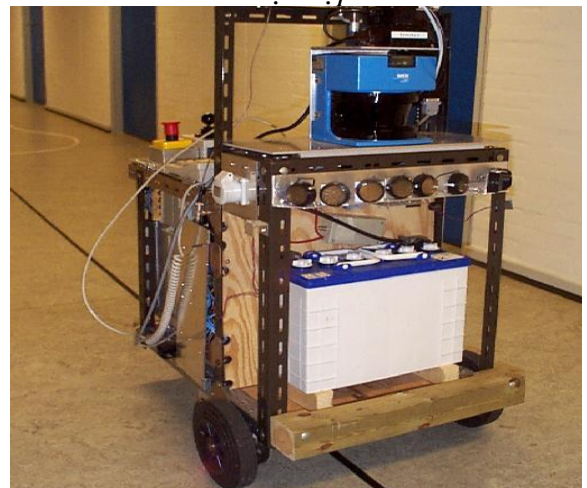


a) Robot Sojourner được sử dụng thám hiểm sao Hỏa năm 1997

b) Robot Khepera dùng để nghiên cứu và học tập.



c) Robot Mbari's Altex Auv hoạt động dưới đáy biển sâu ở Bắc Cực



d) Robot dẫn đường được sử dụng trong các bệnh viện

Hình 1.5. Robot tự hành di chuyển bằng bánh xe

Việc chọn lựa cơ cấu bánh xe là cần thiết và quan trọng, vì ảnh hưởng đến nguồn cung cấp năng lượng chung của robot.

Ưu điểm của robot di chuyển bằng bánh xe là chi phí trung bình thấp hơn so với các loại robot truyền thống; thiết kế và cơ cấu đơn giản; dễ dàng cải tiến, nâng cấp, phát triển và mở rộng;

Tuy nhiên, vì bánh xe có kích thước cố định, không linh hoạt nên diện tích robot tiếp xúc thường nhỏ và bị giới hạn; khả năng kéo và di chuyển yếu.

1.1.3.3. Một số loại robot khác

Hiện nay, tùy theo từng đặc tính và nhu cầu thiết kế, các nhà thiết kế còn nghiên cứu chế tạo ra một số loại robot mang tính đặc thù:

Robot lai (*Hybrid robots*) [75] là hệ thống robot kết hợp nhiều loại phương thức vận động hoặc thao tác để thực hiện nhiệm vụ trong các môi trường đa dạng, phức tạp bởi tính linh hoạt và khả năng thích ứng của chúng. Loại robot này kết hợp các chế độ vận động hoặc thao tác khác nhau nhằm khắc phục những hạn chế về tính di động, đơn chức năng của các loại robot tự hành khác. Chúng được ứng dụng nhiều trong tìm kiếm và cứu nạn, thăm dò, giám sát và hoạt động công nghiệp, nơi yêu cầu đa chức năng với môi trường phức tạp.

Một số loại robot lai thường thấy như: robot có bánh xe có chân là những robot kết hợp cả khả năng di chuyển bằng chân và có bánh xe. Chúng có thể đi bộ hoặc leo bằng chân đối với địa hình gồ ghề hoặc chuyển sang bánh xe để di chuyển nhanh hơn trên các bề mặt bằng phẳng; robot trên không [38] hoạt động ở những địa điểm không thể tiếp cận trên không và sau đó hạ cánh để thực hiện các nhiệm vụ trên mặt đất; robot hình người được kết hợp chuyển động của chân để di chuyển và các cánh tay, bàn tay để thao tác và tương tác như các robot viễn thám, các robot thám hiểm đáy biển [26]; robot dưới nước-trên không [77] có khả năng hoạt động cả dưới nước và trên không để thực hiện các nhiệm vụ dưới nước, sau đó nổi lên và di chuyển đến địa điểm cần thực hiện nhiệm vụ trên không hoặc trên mặt đất.

Robot rắn (*Snake-like robots*) bắt chước chuyển động của rắn bằng cách sử dụng các cơ thể module hoặc phân đoạn. Các robot này hoạt động ở các môi trường phức tạp, không gian hẹp hoặc địa hình không bằng phẳng, bằng cách trườn hoặc

quần quanh các vật thể. Ưu điểm chính của những robot này là chúng rất linh hoạt, đạt được các hành vi không bị giới hạn trong việc leo trèo, bò và bơi [39, 45].

Robot sâu (*Worm-like robots*) là các loại robot được thiết kế với hình dạng và cách di chuyển tương tự như sâu trùn. Chúng có cấu trúc linh hoạt và thường được trang bị các khớp nối hoặc các cấu trúc linh hoạt, sử dụng phương pháp di chuyển bằng sự co bóp nhịp nhàng, co giãn và kéo dài cơ thể giống loại sâu trùn[11]. Loại robot này có khả năng di chuyển mượt mà trong không gian hạn chế, đi qua các khe hẹp và vượt qua các rào cản và được ứng dụng trong khám phá môi trường khó tiếp cận, công nghiệp và xây dựng, y tế, công nghệ địa chất...

Nanorobots [76] là các robot có kích thước ở mức độ nanomet, được tạo ra từ các thành phần nhỏ gọn và được điều khiển để thực hiện các nhiệm vụ cụ thể trong quy mô kích thước nano. Những robot này có khả năng tương tác chính xác với các đối tượng và môi trường cỡ nanomet, cho phép các ứng dụng đa dạng trong lĩnh vực y tế, khoa học, công nghệ và nhiều lĩnh vực khác.

1.1.4. Các ứng dụng và thách thức của robot tự hành

Những năm gần đây, cùng với sự phát triển của công nghệ tự động hóa, của công nghệ robot thì các robot tự hành đã trở nên phổ biến bởi tính ứng dụng của chúng trong các lĩnh vực của đời sống con người. Các robot tự hành trở nên hữu ích hơn nhờ ứng dụng rất đa dạng, từ tự động hoá sản xuất, dịch vụ tư vấn, vận chuyển, dọn dẹp, y tế, nông nghiệp, đến khám phá không gian và các môi trường nguy hiểm như cháy rừng hay thiên tai. Robot tự hành đang được phát triển nhanh chóng và được xem là một trong những công nghệ tiên tiến nhất hiện nay.

Các robot này ít hoặc không chịu sự can thiệp, điều khiển của con người vào quá trình chuyển động, nhận biết và điều hướng trong khi làm việc, cho dù đó là trong môi trường trong nhà hay ngoài trời. Thực tế cho thấy, đối với môi trường trong nhà, robot sử dụng các công nghệ và cảm biến để di chuyển và định vị và giải pháp điều hướng cụ thể sẽ phụ thuộc vào công nghệ và thiết kế của từng loại robot tự hành. Các phương pháp và công nghệ thường được sử dụng là: sử dụng cảm biến (cảm biến hồng ngoại, cảm biến khoảng cách LIDAR hoặc ultrasonic) để đo khoảng cách, để phát hiện vật cản trong môi trường làm việc, từ đó xây dựng bản đồ môi trường, hoặc

áp dụng các kĩ thuật tránh vật cản cho robot tự động thực hiện; cao hơn là việc sử dụng camera để thu thập thông tin hình ảnh từ môi trường, từ đó ứng dụng các kĩ thuật xử lý, nhận dạng ảnh, kĩ thuật phân vùng ảnh, phân vùng đối tượng và thực hiện việc tránh vật cản hoặc lập đường đi trong không gian. Đối với môi trường ngoài trời, robot đòi hỏi các phương pháp và công nghệ cao hơn như: định vị toàn cầu (GPS) để xác định vị trí của robot trong không gian, sử dụng tín hiệu từ các vệ tinh, để xác định vị trí và sử dụng để điều hướng hay cao hơn là ứng dụng công nghệ trí tuệ nhân tạo để phát triển các thuật toán và hệ thống học máy dựa trên dữ liệu thực tế từ môi trường xung quanh, giúp robot thích nghi và cải thiện khả năng điều hướng[22, 25, 47, 63].

1.1.4.1. Ứng dụng của robot tự hành

Với sự phát triển, tiên bộ của công nghệ, nhu cầu sử dụng robot tự hành ngày càng gia tăng bởi tính ứng dụng và hiệu quả mà chúng mang lại [2], bởi thế nhiều nghiên cứu khác nhau được thực hiện nhằm đánh giá sự quan trọng của robot tự hành.

Hiện nay, robot tự hành được ứng dụng rộng rãi trong hầu khắp trong các lĩnh vực của đời sống xã hội (Hình 1.6). Bao gồm: vận tải, hỗ trợ khách hàng, tìm kiếm và cứu hộ, dịch vụ vệ sinh, giám sát, nghiên cứu và giáo dục...

Trong cuộc sống thường ngày của con người, các robot gia đình giúp giám sát an ninh, chăm sóc sức khỏe và giải trí,... [34, 60, 70]. Chúng được lắp đặt trong nhà, kết nối với mạng Wi-Fi hoặc hệ thống thông minh để giám sát, quản lí các hoạt động đang diễn ra, ngoài ra các robot này còn giúp các công việc nhà như vệ sinh nhà cửa, tìm kiếm vật dụng có kết nối, tưới cây, bê dọn các vật dụng nặng... khi cần thiết. Gần đây, các robot gia đình đã được bổ sung, trang bị thêm hệ thống cảnh báo theo thời gian thực, phản ứng tức thì nhằm mục tiêu quản lí, giám sát và chăm sóc người già, trẻ em trong các tình huống cần sự hỗ trợ khẩn cấp [7].

Ở lĩnh vực nghiên cứu và giáo dục, các robot mang lại nhiều lợi ích và ứng dụng nhằm hỗ trợ người dùng phát triển kĩ năng thực hành, giáo dục và nhận thức như: giáo dục STEM cung cấp công cụ hấp dẫn để giảng dạy các khái niệm khoa học, công nghệ, kĩ thuật và toán học; cung cấp trải nghiệm học tập thực hành, giúp sinh viên áp dụng kiến thức lí thuyết vào các tình huống, thúc đẩy sự sáng tạo và tư duy phản biện; nghiên cứu tương tác giữa người và robot là các nghiên cứu liên quan đến

tương tác giữa người và robot (HRI) hay hỗ trợ giáo dục đặc biệt là việc ứng dụng robot trong công tác hỗ trợ sinh viên đặc biệt, khuyết tật học tập, nghiên cứu thúc đẩy tính hòa nhập trong môi trường giáo dục. Hầu hết các tiếp cận này kích thích sự quan tâm đối với nghiên cứu nhằm hỗ trợ và củng cố kiến thức và kỹ năng đối với không chỉ học sinh, sinh viên mà còn cho mọi giới, mọi lứa tuổi. Các nghiên cứu chế tạo, ứng dụng robot tự hành không chỉ gia tăng ở môi trường đại học và viện nghiên cứu, mà còn ở các cấp học thấp như trung học và tiểu học [21].



Hình 1.6. Ứng dụng của robot tự hành [2]

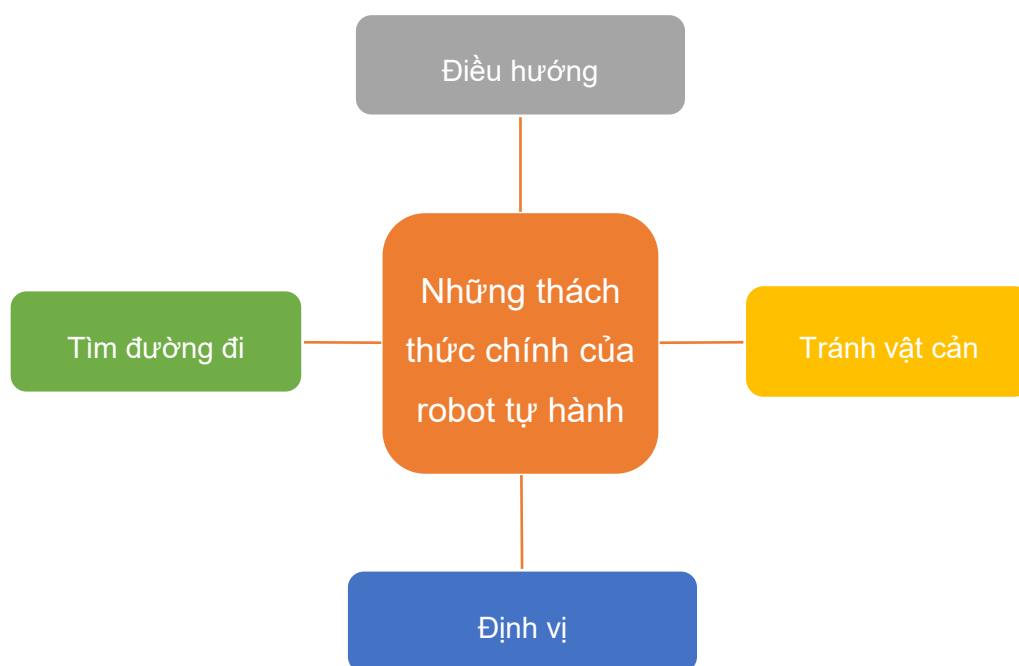
Robot tự hành cũng thu hút được nhiều sự quan tâm trong công nghiệp khai thác mỏ [20] bởi hiệu quả khai thác, tăng tính an toàn cho thợ mỏ mà chúng mang lại. Các robot này được ứng dụng để theo dõi, giám sát con người và máy móc trong môi trường hầm mỏ. Nền tảng robot tự hành được kết hợp với hệ thống các cảm biến khoảng cách, cảm biến hình ảnh nhiệt và hệ thống âm thanh, được tổng hợp và điều khiển bằng mạng neuro. Chúng giúp cho robot di chuyển vào những khu vực hầm mỏ, hầm lò nơi con người không thể vào được hoặc có xác định khu vực tiềm tàng nguy cơ gây nguy hiểm cho con người.

Ngoài ra, robot tự hành còn được ứng dụng ở nhiều lĩnh vực khác như chữa cháy, nông nghiệp, hướng dẫn viện bảo tàng và thư viện, khám phá hành tinh, tuần

tra, tìm hiểu, ứng dụng hóa đầu cũng như cho cả ứng dụng trong gia đình và công nghiệp [75],...

1.1.4.2. Thách thức đặt ra với robot tự hành

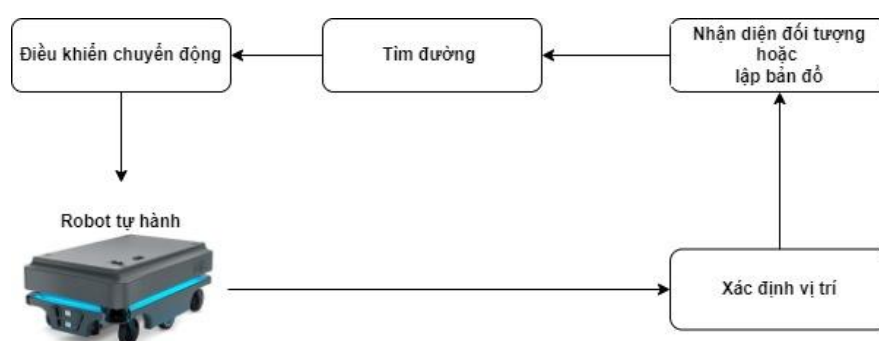
Vai trò và ứng dụng của robot tự hành là không hề nhỏ trong mọi lĩnh vực của đời sống của con người. Tuy nhiên, chúng vẫn chưa đạt được hiệu suất tối ưu bởi lẽ yêu cầu ngày càng cao của con người cũng như sự phát triển của các công nghệ liên quan như công nghệ cơ điện tử, điều khiển, cảm biến, trí tuệ nhân tạo. Bởi vậy, hiện nay robot tự hành đang phải đối mặt với những thách thức căn bản, phổ biến là: điều hướng, tìm đường đi, định vị và tránh vật cản. Những thách thức này (Hình 1.7) đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong thời gian gần đây.



Hình 1.7. Những thách thức của robot tự hành [2]

Tìm đường đi là thách thức chính đối với robot tự hành, tập trung giải quyết tìm đường đi hiệu quả cho robot tự hành. Bao gồm: tạo bản đồ, nhận biết môi trường và xây dựng quỹ đạo di chuyển tốt nhất cho robot để đạt mục tiêu. Những vấn đề này hướng đến việc nghiên cứu và phát triển các thuật toán mạnh và tích hợp cảm biến để xử lý các tình huống động trong môi trường làm việc không biết trước.

Điều hướng là quá trình xác định chính xác vị trí của robot, điều khiển, chuyển hướng di chuyển của robot trong quá trình tìm đường, tránh vật cản (Hình 1.8). Quá trình điều hướng đảm bảo cho robot di chuyển hợp lý và tối ưu theo tiêu chí thời gian hoặc quãng đường theo một quỹ đạo nào đó, đồng thời đảm bảo không xảy ra va chạm trong quá trình di chuyển [27].



Hình 1.8. Vấn đề điều hướng của robot tự hành

Định vị là yếu tố quan trọng để xác định vị trí của robot, điểm đích cần hướng đến và các vật cản (nếu có) trong môi trường hoạt động. Các kỹ thuật đang được sử dụng để định vị cho robot tự hành như: phương pháp ứng dụng GPS với môi trường hoạt động rộng không đòi hỏi tính chính xác cao; phương pháp ứng dụng kỹ thuật kết hợp cảm biến, camera, lidar hoặc cảm biến trọng lực để ước tính vị trí một cách chính xác đối với môi trường hoạt động hẹp, độ phức tạp cao đòi hỏi tính chính xác. Dựa trên yêu cầu thực tế, và tính ứng dụng của robot tự hành về không gian hoạt động hẹp, có sự tham gia của các thành phần khác như con người, robot cộng tác, vật cản, nên phương pháp ứng dụng GPS không được coi là lựa chọn hiệu quả. Thay vào đó, các thuật toán tránh va chạm được cài đặt dựa trên ứng dụng kỹ thuật kết hợp: cảm biến, camera, lidar hoặc cảm biến.

Tránh vật cản là một thách thức quan trọng, yêu cầu các robot phải nhận biết và phân tích dữ liệu cảm biến để phát hiện và phân loại các vật cản một cách chính xác. Trên cơ sở tính toán và áp dụng các thuật toán tránh vật cản, robot ra quyết định lựa chọn cách tránh hoặc điều hướng xung quanh vật cản mà vẫn bám sát quỹ đạo

yêu cầu, theo thời gian thực. Cho đến nay, vấn đề tránh vật cản không còn tập trung chủ yếu trong môi trường tĩnh, mà còn được quan tâm giải quyết trong không gian tự do, môi trường không xác định với yêu cầu robot tự hành không va chạm với các vật cản (con người, vật, robot cộng tác) [35].

Bên cạnh những thách thức chủ yếu, robot tự hành còn phải quan tâm đến các vấn đề khác như quản lí năng lượng nhằm tối ưu hóa mức tiêu thụ năng lượng; phối hợp chuyển động của các thành phần cấu tạo (cánh tay, bộ điều khiển); bảo đảm các giao thức truyền thông hiệu quả; phân bổ nhiệm vụ và chiến lược hợp tác giữa robot và các thành phần trong hệ thống (robot cộng tác, thiết bị điều khiển,...).

Vì vậy, những thách thức này tạo nên sự thu hút và mối quan tâm rất lớn cho các nhà thiết kế, các nhà nghiên cứu trong việc phát triển các giải pháp, công nghệ để cải tiến, nâng cao hiệu quả của robot tự hành.

1.2. Một số tiếp cận về tránh va chạm

Vấn đề phát hiện, tránh va chạm đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khoa học, tiêu biểu nhất, mới mẻ nhất phải kể đến lĩnh vực công nghệ thực tại ảo với nhiều ứng dụng trong khoa học, công nghệ, đời sống xã hội... Và vấn đề phát hiện, tính toán, tránh va chạm hay xử lí va chạm cũng là một thách thức đối với lĩnh vực này để xây dựng các ứng dụng mô phỏng trước khi tiến hành triển khai thực tế đảm bảo tính hiệu quả, đúng đắn của phương pháp, kĩ thuật. Hướng tiếp cận này nhằm thay thế cho các thí nghiệm, thực nghiệm trong thực tế, giảm thiểu chi phí và đảm bảo an toàn cho môi trường. Nhóm nghiên cứu thuộc Viện Công nghệ Thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam đã tiếp cận và nghiên cứu các vấn đề về tránh va chạm trong thực tại ảo như: tính toán va chạm [82, 87], phát hiện, tránh va chạm và ứng dụng trong nhiều vấn đề thực tế [87, 88] hay xử lí sau va chạm [81].

Đối với lĩnh vực công nghệ robot nói chung và robot tự hành nói riêng, các nghiên cứu về phát hiện và tránh va chạm cho robot đã được thực hiện từ những năm 1970 và 1980, khi các nhà khoa học sử dụng cảm biến và thuật toán để giúp robot phát hiện và tránh va chạm với các vật cản trong môi trường. Cùng với sự phát triển

của công nghệ robot, yêu cầu về hệ thống cảm biến, các thuật toán phát hiện, cảnh báo và tránh va chạm cho robot đã và đang được nghiên cứu hướng đến sự chính xác cao, đảm bảo hoạt động của robot gần nhất đối với các hoạt động của con người. Các nghiên cứu gần đây tiếp cận với kỹ thuật cảm biến LiDAR, cảm biến quang học, nhận dạng và xử lý hình ảnh kết hợp với kỹ thuật máy học, mạng neuro hay trí tuệ nhân tạo nhằm trang bị cho robot khả năng phát hiện và tránh va chạm với các đối tượng trong môi trường, đảm bảo robot hoạt động an toàn.

Cùng với việc tìm đường thì vấn đề tránh va chạm hay cảnh báo va chạm cho các robot tự hành với các vật cản đang là một vấn đề quan trọng được đặt ra đối với các nhà nghiên cứu, các nhà sản xuất hay các nhà tư vấn giải pháp tự động hóa. Bởi vậy, những năm gần đây các nghiên cứu về phát hiện và tránh va chạm cho robot tự hành đã thu hút sự quan tâm của nhiều nhà khoa học, nhà nghiên cứu nhằm giải quyết thách thức về tránh vật cản cho các robot [59]. Các hướng nghiên cứu này không chỉ giải quyết vấn đề với robot tự hành hoạt động đơn lẻ trong một môi trường tĩnh cố định, mà còn được mở rộng với nhóm các robot cùng hoạt động đồng thời trong môi trường động có chứa các chướng ngại vật chuyển động.

Các nghiên cứu về thuật toán tránh va chạm được chia thành hai loại [1, 71] gồm: *Tránh va chạm dựa trên phản ứng* (các thuật toán theo thời gian thực, trong môi trường động và các robot tính toán, xác định và lựa chọn phương thức di chuyển đảm bảo tránh vật cản và hướng đích); *Tránh va chạm dựa trên kế hoạch* (trên cơ sở sử dụng các thông tin sẵn có của môi trường hoạt động, các robot tính toán để tạo lập quỹ đạo, lập kế hoạch tránh vật cản theo quỹ đạo, lộ trình và đường đi cho trước).

1.2.1. Tránh va chạm cho robot tự hành dựa trên lập kế hoạch

Tránh va chạm dựa trên việc lập kế hoạch đường đi cho robot nhằm mục tiêu giúp robot điều hướng cách an toàn và hiệu quả trong môi trường không xác định hoặc biết một phần. Có nhiều phương pháp đã và đang được nghiên cứu:

Thuật toán Dijkstra được Edsger Wybe Dijkstra đã giới thiệu thuật toán Dijkstra vào năm 1959 [23]. Phương pháp này mô hình hóa môi trường hoạt động của robot thành bản đồ có cấu trúc liên kết, là một tập hợp các nút trên bản đồ topo

của không gian hoạt động. Việc tìm đường đi cho robot được hiểu như là việc tìm đường dẫn ngắn nhất giữa một nút và một nút khác trong bản đồ có hướng để tìm đường đi tối ưu toàn cục cho robot [33, 71]. Phương pháp này giải quyết được vấn đề tìm đường cho robot và luôn đảm bảo khoảng cách từ điểm xuất phát đến điểm hiện tại là nhỏ nhất. Tuy nhiên, phương pháp này chỉ phù hợp cho mô hình ngoại tuyến và môi trường hoạt động của robot là tĩnh.

*Thuật toán A** [1, 58] sử dụng phép đánh giá ước tính khoảng cách từ một điểm đến điểm đích và sử dụng phép đánh giá này với chi phí đã thực hiện để chọn đường đi tiếp theo. Thứ tự ưu tiên chọn điểm tiếp theo được xác định bởi tổng chi phí đã di chuyển và ước tính khoảng cách đến đích. Thứ tự ưu tiên này cho phép A* tìm được đường đi ngắn nhất giữa hai điểm trong bản đồ của robot. Cả Dijkstra và A* đều phụ thuộc vào việc xây dựng thông tin bản đồ cho trước cho robot. Tuy nhiên, môi trường và không gian làm việc luôn đòi hỏi sự thay đổi thiết kế theo nhu cầu nên việc thay đổi, cập nhật hệ thống bản đồ, sơ đồ cũng phải thay đổi theo. Do vậy, các phương pháp này sẽ phải thay đổi dữ liệu đầu vào để đảm bảo tính kế thừa, cập nhật.

Phương pháp Model Predictive Control (MPC) [57, 79] tham chiếu đến đường dẫn được thuật toán A* tạo trước. Phương pháp này cũng đòi hỏi phải tạo bản đồ mỗi khi thay đổi bố cục không gian làm việc. Shini và cộng sự [37] đã đề xuất phương pháp quy hoạch chỉ đường cho nhiều robot chỉ sử dụng MPC linh hoạt hơn, đáp ứng với những thay đổi của không gian làm việc của robot. Trong đó, các hàm mục tiêu MPC được cập nhật nhằm giải quyết các vấn đề khi hàm MPC hoạt động độc lập hoặc robot không thể di chuyển khi không có sự tối ưu cục bộ. Các nghiên cứu trên đều dựa vào bản đồ được xây dựng trước, mà thực tế trong tìm đường, việc phát sinh các tình huống với các vật cản (robot khác, con người, máy móc) là không thể tránh khỏi. Bởi vậy, ứng với bài toán thực tế khi robot hoạt động trong môi trường động thì phương pháp này còn hạn chế.

Phương pháp Rapidly-exploring Random Tree (RRT) [54, 71] dựa trên thuật toán lập kế hoạch chuyển động xác suất được sử dụng để tìm đường đi giữa điểm bắt đầu và mục tiêu trong không gian nhiều chiều. Thuật toán RRT tạo ra một cây bao phủ toàn bộ không gian bản đồ của robot bằng cách tăng dần cấu trúc dạng cây (lấy

mẫu ngẫu nhiên các điểm trong môi trường và kết nối với nút gần nhất trong cây cho đến khi đến điểm đích). Tuy nhiên, thuật toán này có hạn chế về độ phức tạp và thời gian tính toán khi vùng hoạt động của robot có nhiều vật cản, theo thời gian thực.

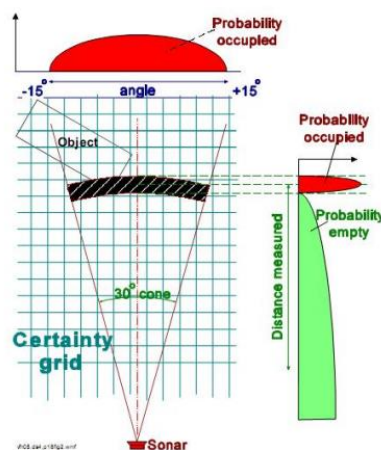
Phương pháp Simultaneous Localization and Mapping (SLAM) là một phương pháp dựa trên bản đồ của môi trường mà robot đang di chuyển. Bản đồ này được thực hiện bằng cách sử dụng các cảm biến như camera, lidar hoặc GPS để quan sát và ghi lại môi trường, đồng thời sử dụng thuật toán để tổng hợp dữ liệu để xử lý bản đồ và xác định vị trí. Chen và cộng sự [16] đã đề xuất, phát triển phương pháp này để tìm đường cho robot bằng cách tự động cập nhật tọa độ bản đồ. Tuy nhiên, để tạo ra bản đồ thực tế cần phải lập thủ công các nút chính nên dẫn đến việc tiêu tốn nhiều tài nguyên phục vụ tính toán.

1.2.2. Tránh va chạm cho robot tự hành dựa trên phản ứng

Hiện nay, có nhiều công trình nghiên cứu tránh va chạm dựa trên phản ứng đã được đề xuất và thực hiện như: Certainty Grid [62], Artificial Potential Field [36, 42, 52], Virtual Force Field [9, 10, 46]... các phương pháp này được áp dụng phổ biến cho robot di động, nhưng chưa xét đến giới hạn động. Bởi vậy, trong nhiều trường hợp khi robot di chuyển trong một không gian có dạng cổ chai, chúng sẽ không tìm được lối thoát. Các phương pháp này tập trung vào việc phát hiện các vật cản (thông qua hệ thống cảm biến), tính toán khoảng cách, vị trí của vật cản, áp dụng các thuật toán để tính toán vị trí và thay đổi quỹ đạo của robot theo thời gian thực.

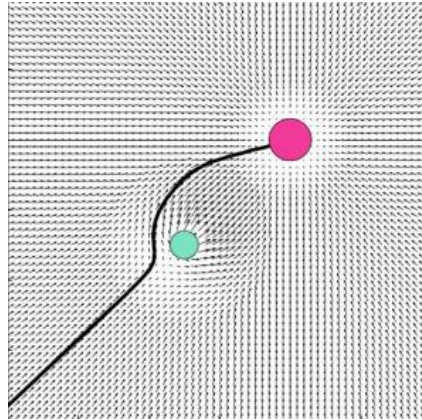
Phương pháp chia lưới xác định được hướng đến biểu diễn các vật cản trong một mô hình môi trường dạng lưới (grid-type world model) [17, 55] được dùng để mô phỏng và điều hướng robot trong các môi trường rời rạc. Mỗi ô trong lưới đại diện cho một trạng thái hoặc điều kiện cụ thể của môi trường. Trạng thái này có thể bao gồm các thuộc tính khác nhau như trạng thái chiếm đầy, loại địa hình, sự hiện diện của các vật cản. Trạng thái của mỗi ô được cập nhật dựa trên việc sử dụng dữ liệu cảm biến để xác định xem mỗi ô có bị chiếm dụng hay không. Từ đó, kiểm tra trạng thái lưới để xác định xem robot có xảy ra va chạm, đồng thời đưa ra đề xuất điều chỉnh đường đi.

Mở rộng phương pháp chia lưới, Moravec và cộng sự [61, 62] đề xuất mô hình lưới xác định gọi là “*Certainty Grid*” [8] phù hợp với những môi trường nhiễu cao với dữ liệu từ cảm biến siêu âm. Trong phương pháp này, vùng hoạt động của robot được biểu diễn bằng một mảng hai chiều các phần tử hình vuông, gọi là các cell. Mỗi cell chứa một giá trị xác định gọi là CV để đo mức độ đáng tin cậy việc xuất hiện vật cản có trong cell đó. Với phương pháp này, CV được cập nhật bằng một hàm xác suất sử dụng cảm biến siêu âm có một tầm quan sát hình nón, dữ liệu trả về là một giá trị đo hướng tâm khoảng cách từ robot đến vật cản gần nhất nằm trong vùng hình nón đó. (Hình 1.9). Từ đó, giá trị CV của mỗi cell được gia tăng và việc đánh giá sự xuất hiện, của vật cản dựa trên giá trị của bảng 2 chiều với các giá trị CV tương ứng nằm trong các cell.



Hình 1.9. Phương pháp chia lưới xác định [8]

Phương pháp trường thế năng nhân tạo (Artificial Potential Field) được đề xuất bởi O. Khatib và cộng sự [19, 42]. Các tác giả đã nghiên cứu, giả định rằng robot di chuyển trong một trường thế năng nhân tạo được tạo ra bởi các véc tơ lực phản, lực ngẫu nhiên, lực hấp dẫn được tạo bởi các vật cản và điểm đích. Trường thế năng nhân tạo là véc tơ lực tổng hợp của các lực tạo bởi vật cản (lực phản) và lực hướng đích (lực kéo) (Hình 1.10). Việc tính toán vị trí mới của robot được thông qua véc tơ tổng hợp lực, robot sẽ điều hướng và tiếp tục thực hiện quá trình đó cho đến khi đến được vị trí đích.



Hình 1.10. Phương pháp trường thế năng nhân tạo

Mở rộng đề xuất của O. Khatib về lí thuyết trường lực nhân tạo, một số tác giả đã phát triển và ứng dụng lí thuyết này để giải quyết các bài toán tránh va chạm cho robot. Krogh và cộng sự [48] mở rộng nghiên cứu bằng cách xem xét thêm tác động của tốc độ của robot ở trạng thái gần với vật cản; Krogh và Thorpe [49] tiếp cận trường tiềm năng tổng quát (GPF) là một phần mở rộng của phương pháp trường tiềm năng cho việc lập quỹ đạo chuyển động cho robot bằng cách kết hợp ảnh hưởng của tất cả các lực kèm theo trọng số quan trọng của mỗi thành phần lực để xác định tác động của mỗi lực đối với việc điều hướng của robot. Tuy nhiên, GPF có hạn chế là tìm ra các trọng số phù hợp với từng thành phần lực để đạt được hành vi điều hướng mong muốn mà không bị kẹt hoặc gặp cực tiểu cục bộ trong trường tiềm năng; Newman và Hogan [66] cũng giới thiệu cách xây dựng hàm tiềm năng thông qua việc kết hợp các đặc tính của vật cản.

Hầu hết các tiếp cận trên đều có những cải tiến đáng kể, được kiểm chứng bằng mô hình toán học và mô phỏng nhưng không được cài đặt thực tế cho robot với các dữ liệu cảm biến. Bởi vậy, Brooks [12, 13] và Arkin [6] đã tiến hành cài đặt phương pháp này cho một robot sử dụng hệ thống cảm biến siêu âm. Brooks cài đặt và kiểm thử phương pháp này cho các robot bằng cách đọc từng tín hiệu của cảm biến siêu âm và tính toán các véc tơ lực đẩy của các vật cản. Khi tổng hợp véc tơ lực đẩy, nếu lực đó vượt qua ngưỡng nhất định nào đó thì robot dừng lại, hoặc không đạt hiệu quả trong vấn đề tránh vật cản. Như vậy, các phương pháp này chủ yếu được thực

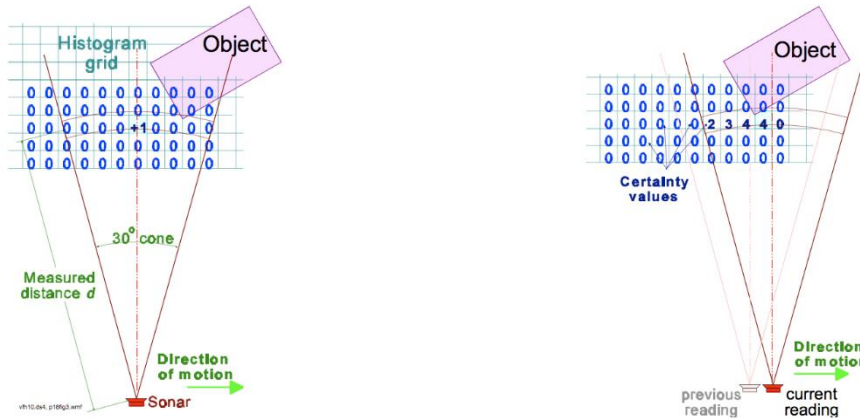
hiện trong mô hình giả định, thông qua hình dạng hình học của các vật cản để tạo ra đường đi của robot theo hình thức không trực tuyến (*offline*).

Ở Việt Nam, một số nghiên cứu về tránh vật cản cho robot đã được thực hiện như: nhóm tác giả thuộc trường Đại học Sư phạm Kỹ thuật thành phố Hồ Chí Minh [83] đã nghiên cứu phương pháp trường thế năng để xây dựng thuật toán tránh vật cản cho robot di động; nhóm nghiên cứu thuộc trường Đại học Công nghệ, Đại học Quốc gia Hà Nội [85] đã nghiên cứu ứng dụng phương pháp trường thế nhân tạo để giải quyết bài toán tránh va chạm trong hệ thống đa robot bằng cách đề xuất cấu trúc trường thế và luật điều khiển dựa trên vectơ tổng hợp lực thế và lực kéo trên mỗi robot. Các mô phỏng và thực nghiệm cho thấy phương pháp này giải quyết được bài toán tránh va chạm cho hệ thống robot với kết quả tốt, độ ổn định cao.

Phương pháp trường lực ảo (Virtual Force Field) được đề xuất bởi Borenstein và cộng sự [9, 10] dựa trên khái niệm tạo ra một trường lực ảo xung quanh robot và gây ra những ảnh hưởng đến chuyển động của robot. Phương pháp này sớm được cài đặt, áp dụng cho bài toán tránh va chạm cho robot và đáp ứng các yêu cầu về thời gian thực, tốc độ cao, cho phép robot có thể di chuyển nhanh, liên tục và chuyển động trơn tru hơn khi gặp vật cản trong môi trường mà không phải dừng lại.

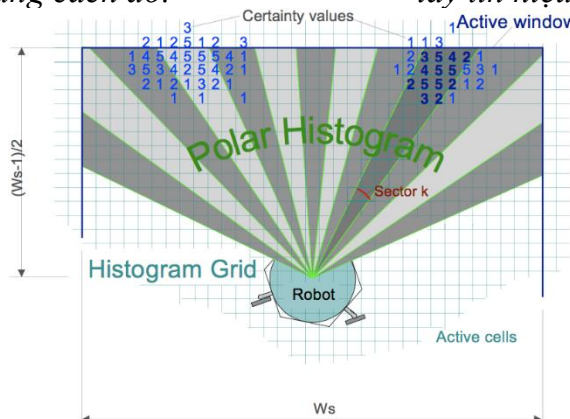
Trong VFF, môi trường được thể hiện dưới dạng lưới hoặc không gian liên tục và vị trí hiện tại của robot được xác định, lực đẩy được gán cho từng chương ngại vật và có hướng đẩy robot ra xa vật cản với độ lớn tỉ lệ nghịch với khoảng cách giữa robot và vật cản và một lực hút tạo bởi vị trí mục tiêu kéo robot đến đích mong muốn có độ lớn tỉ lệ thuận với khoảng cách giữa robot và mục tiêu. Để đo giá trị lực, các tác giả sử dụng một mạng lưới hai chiều biểu đồ Descartes C để biểu diễn vật cản giống như phương pháp lưới xác định, trong đó mỗi Cell (i,j) trong biểu đồ lưới là giá trị chắc chắn xác định tồn tại của các vật cản tại vị trí đó. Cụ thể, giá trị các Cell(i,j) được cập nhật bằng cách đọc tín hiệu từ hệ thống cảm biến theo thời gian thực và phụ thuộc vào hướng di chuyển của robot. Mặt khác, giá trị độ tin cậy ở các Cell (i,j) sẽ được tăng thêm khi tại thời điểm cảm biến báo về việc gặp vật cản. Kết quả sau cùng ta sẽ xây dựng được một biểu đồ biểu diễn sự xuất hiện của các vật cản

trên biểu đồ lưới (Hình 1.11). Việc tính toán và cập nhật các giá trị Cell(i, j) đơn giản thông qua việc thu nhận liên tục và theo thời gian thực đối với mỗi cảm biến siêu âm.



a. Chỉ một ô được tăng giá trị cho mỗi lần đọc tín hiệu, giá trị trả về tương ứng với khoảng cách đo.

b. Phân phối xác suất biểu đồ thu được bằng các tín hiệu liên tục và lấy tín hiệu từ cảm biến.



Hình 1.11. Phương pháp trường lực ảo (VFF) [10]

Phương pháp này không hướng đến giải quyết các bài toán tránh va chạm cho robot ở môi trường động, thuần túy mới chỉ giải quyết trong môi trường tĩnh cố định. Do vậy, robot có thể không đi được trong trường hợp di chuyển khu vực nút cổ chai, hoặc không tránh được vật cản.

Phương pháp *Elastic strips* do Khatib đề xuất [19, 40, 41] giả định quỹ đạo của robot là một dải đàn hồi, thuật toán tránh va chạm dựa trên dải đàn hồi đã được thử nghiệm và cho thấy phù hợp với các hệ thống thời gian thực.

1.3. Kết luận và vấn đề nghiên cứu

Vấn đề *tránh va chạm* đã và đang là một thách thức chủ yếu đối với các robot tự hành trong quá trình hoạt động. Bởi vậy, luận án này hướng đến nghiên cứu và phát triển một số kỹ thuật tránh va chạm cho robot tự hành để đảm bảo tính hiệu quả, ổn định và theo thời gian thực cho các robot tự hành. Trong đó, luận án tiếp cận nghiên cứu theo hướng *tránh va chạm dựa trên phản ứng* bởi lẽ đây đang là một vấn đề thách thức đối với các robot tự hành, khi mà môi trường hoạt động không còn đơn thuần được xác định trước, lập sẵn kế hoạch mà hiện nay môi trường hoạt động của các robot phức tạp hơn, đa dạng, nhiều thành phần tham gia bao gồm các vật cản, người vận hành và các robot hoạt động song song, và các thành phần này đều di chuyển theo một quỹ đạo riêng, không biết trước. Cùng với đó là yếu tố thời gian thực cũng được yêu cầu để đáp ứng ngày càng cao các ứng dụng của robot tự hành.

Cụ thể:

1- Nghiên cứu và phát triển thuật toán tránh va chạm cho robot dựa trên cấu trúc cây phân lớp hệ bao BVH kết hợp với kỹ thuật Elastic strips.

2- Nghiên cứu và đề xuất thuật toán tránh va chạm cho robot dựa trên tính toán xác suất và phân vùng đồng mức xác suất va chạm cho robot tự hành.

CHƯƠNG 2. KỸ THUẬT TRÁNH VA CHẠM DỰA TRÊN CẤU TRÚC CÂY PHÂN LỚP HỆ BAO BVH KẾT HỢP THUẬT TOÁN ELASTIC STRIPS

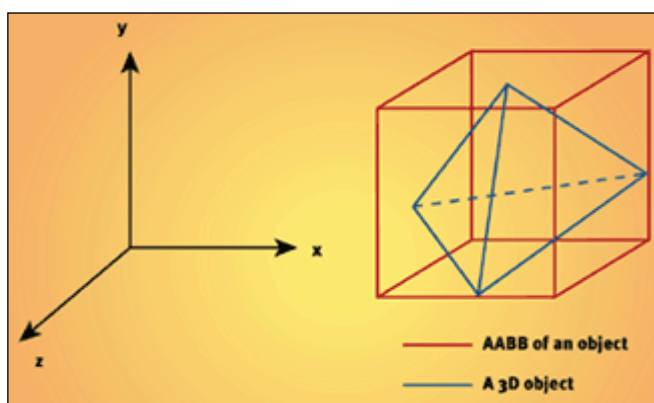
Chương 2 trình bày về kỹ thuật tránh va chạm cho robot tự hành dựa trên cải tiến phương pháp Elastic strips sử dụng kỹ thuật cấu trúc cây phân lớp hệ bao BVH.

2.1. Phương pháp phát hiện va chạm sử dụng hộp bao và cấu trúc hệ bao

Phương pháp phát hiện va chạm dựa trên hộp bao [24, 25, 87] là một phương pháp phổ biến để xác định hai đối tượng có va chạm hay không thông qua kết quả kiểm tra nhanh khả năng va chạm của hai hộp bao được của hai đối tượng đó. Các dạng hộp bao của các đối tượng có thể là hộp bao dạng theo trục (AABB), hộp bao hình tròn, hộp bao hình ellipse hay hộp bao theo hướng (OBB). Tuy nhiên, phổ biến hơn cả và thường được sử dụng nhiều là hộp bao theo trục và hộp bao theo hướng.

2.1.1. Phát hiện, tính toán va chạm sử dụng hộp bao dạng trục

AABB là hộp bao có dạng hình hộp chữ nhật có các cạnh (trục) song song với các trục tọa độ tương ứng và bao lấy đối tượng (Hình 2.3).



Hình 2.1. Hộp bao AABB của đối tượng [24]

Hộp bao AABB bao gồm một tâm C , ba hệ số a_0, a_1, a_2 tương ứng là độ dài theo ba trục tọa độ của hình hộp.

Cho hai hộp bao AABB xác định bởi (C_1, a_0, a_1, a_2) và (C_2, b_0, b_1, b_2) với giả sử $a_i > 0, b_j > 0, i, j = 0, 1, 2$. Để kiểm tra va chạm, ta xác định tọa độ cao nhất và thấp

nhất của mỗi hộp bao. Kí hiệu $(x_{\min 1}, y_{\min 1}, z_{\min 1})$, $(x_{\max 1}, y_{\max 1}, z_{\max 1})$ là toạ độ thấp nhất và cao nhất của hộp bao có tâm C_1 :

$$\begin{aligned}x_{\min 1} &= C_0 - \frac{a_0}{2}, & x_{\max 1} &= C_0 + \frac{a_0}{2} \\y_{\min 1} &= C_0 - \frac{b_0}{2}, & y_{\max 1} &= C_0 + \frac{b_0}{2} \\z_{\min 1} &= C_0 - \frac{c_0}{2}, & z_{\max 1} &= C_0 + \frac{c_0}{2}\end{aligned}\tag{2.1}$$

Tương tự, ta tính được $(x_{\min 2}, y_{\min 2}, z_{\min 2})$, $(x_{\max 2}, y_{\max 2}, z_{\max 2})$ là toạ độ thấp nhất và cao nhất của hộp bao xác định bởi tâm C_2 . Hai hộp bao AABB va chạm nhau nếu xảy ra một trong bốn điều kiện sau:

$$\begin{aligned}(x_{\min 1}, y_{\min 1}, z_{\min 1}) &\in [(x_{\min 2}, y_{\min 2}, z_{\min 2}), (x_{\max 2}, y_{\max 2}, z_{\max 2})] \\(x_{\max 1}, y_{\max 1}, z_{\max 1}) &\in [(x_{\min 2}, y_{\min 2}, z_{\min 2}), (x_{\max 2}, y_{\max 2}, z_{\max 2})] \\(x_{\min 2}, y_{\min 2}, z_{\min 2}) &\in [(x_{\min 1}, y_{\min 1}, z_{\min 1}), (x_{\max 1}, y_{\max 1}, z_{\max 1})] \\(x_{\max 2}, y_{\max 2}, z_{\max 2}) &\in [(x_{\min 1}, y_{\min 1}, z_{\min 1}), (x_{\max 1}, y_{\max 1}, z_{\max 1})]\end{aligned}\tag{2.2}$$

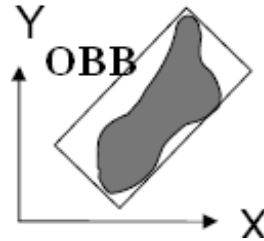
Việc tìm điểm va chạm giữa hai hộp bao dạng trục được xác định và lựa chọn là các đỉnh tương ứng với một trong bốn trường hợp trên. Hộp bao dạng trục này được đánh giá là đơn giản, xây dựng dễ dàng và kiểm tra va chạm cũng không quá phức tạp. Tuy nhiên, chính bởi hộp bao dạng trục này đơn giản, nên tạo ra nhiều khoảng trống giữa đối tượng và hộp bao của nó (đặc biệt: khi đối tượng không nằm song song với các trục toạ độ và có dạng dài thì khoảng trống này càng lớn) nên độ chính xác không cao bằng các dạng hộp bao khác.

Trong thực tế, kĩ thuật hộp bao dạng trục thường được sử dụng để giới hạn vùng va chạm, còn để kiểm tra và tìm điểm va chạm chính xác hơn ta có thể sử dụng kĩ thuật hộp bao theo hướng (OBB).

2.1.2. Phát hiện, tính toán va chạm sử dụng hộp bao theo hướng

Thực chất, hộp bao theo hướng (OBB) là hộp bao dạng trục có hướng bất kì. Loại hộp bao này ưu điểm hơn hộp bao dạng trục là giảm không gian trống giữa đối tượng và hộp bao được tạo ra bao quanh nó, nhưng việc tạo ra và thao tác trên dạng hộp bao loại này phức tạp hơn. Chính bởi ưu điểm về giảm khoảng trống giữa đối

tượng và hộp bao tạo ra, thêm với thao tác không quá phức tạp nên kỹ thuật này thường được dùng nhiều hơn bởi nó có độ chính xác hơn các dạng hộp bao khác, trong phạm vi sai số chấp nhận được (Hình 2.1).



Hình 2.2. Hộp bao OBB của đối tượng [24]

Một hộp bao theo hướng được xác định bởi một tâm C , ba véc tơ $\vec{A}_0, \vec{A}_1, \vec{A}_2$ chỉ hướng của hình hộp và 3 hệ số độ dài tương ứng với kích thước của hình hộp là $a_0 > 0, a_1 > 0, a_2 > 0$. Khi đó, 8 đỉnh của hình hộp sẽ được xác định như sau:

$$C + \sum_{i=0}^2 s_i a_i * \vec{A}_i \quad |s_i| = 1, i = 0, 1, 2. \quad (2.3)$$

Việc xác định va chạm của hai hộp bao theo hướng được chia làm hai mức [24]: mức một là kiểm tra “nhẹ” xem có va chạm nào xảy ra không? Nếu không có va chạm xảy ra thì hệ thống thực hiện công đoạn tiếp theo, ngược lại quá trình sẽ chuyển sang mức hai để tìm chính xác điểm va chạm của hai hộp bao.

Để thực hiện việc kiểm tra nhanh xem có va chạm nào xảy ra ở mức một hay không, thông qua định lý về hai mặt phẳng không giao nhau nếu và chỉ nếu nó được tạo ra bởi một đường thẳng chung trong hình học Euclid, ta thấy rằng nếu hai khối đa diện lồi A và B có giao điểm, thì ta có thể tìm một đường thẳng d trong không gian giữa hai giao điểm đó. Đồng thời, vì hai khối đa diện lồi, nên một mặt phẳng P có thể được tạo ra bởi đường thẳng d này và một điểm trong không gian và mặt phẳng P sẽ chia hai khối A và B ra thành hai phần riêng biệt. Do vậy, hai khối đa diện lồi sẽ không giao nhau nếu có thể cô lập được chúng bằng một mặt phẳng P thỏa mãn một trong hai điều kiện sau:

- P song song với một mặt nào đó của một trong hai khối đa diện.
- P chứa một cạnh thuộc đa diện thứ nhất và một đỉnh thuộc đa diện thứ hai.

Điều kiện cần và đủ để kiểm tra hai khối đa diện lồi có giao nhau hay không là kiểm tra sự giao nhau giữa các hình chiếu của chúng lên đường thẳng vuông góc với mặt phẳng P ở trên và đường thẳng này được gọi là trục cô lập, chính là việc áp dụng lí thuyết này trong việc kiểm tra va chạm giữa hai hộp bao theo hướng (trong điều kiện các hộp bao này là những khối đa diện lồi).

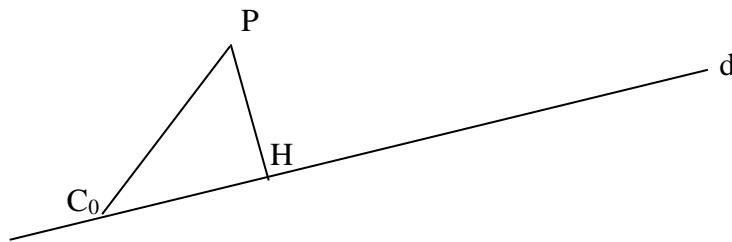
2.1.2.1. Kiểm tra va chạm

Cho hai hình bao OBB xác định bởi các thông số $(C_0, A_0, A_1, A_2, a_0, a_1, a_2)$ và $(C_1, B_0, B_1, B_2, b_0, b_1, b_2)$. Ta thấy rằng các tình huống mà hai hộp bao OBB tiếp xúc với nhau chỉ có thể là một trong 6 trường hợp sau đây: mặt - mặt, mặt - cạnh, mặt - đỉnh, cạnh - cạnh, cạnh - đỉnh, đỉnh - đỉnh. Từ đó, ta xác định được tập ứng cử viên các trục cô lập là:

- 3 trục chỉ hướng của hộp bao thứ nhất (\vec{A}_i)
- 3 trục chỉ hướng của hộp bao thứ hai (\vec{B}_j)
- 9 trục tạo bởi tích có hướng của một trục thuộc hộp bao thứ nhất và một trục thuộc hộp bao thứ hai $(\vec{A}_i \otimes \vec{B}_j)$.

Mặt khác, nếu một trục là trục cô lập thì khi tịnh tiến đến vị trí nào, nó vẫn là trục cô lập. Bởi vậy, không mất tính tổng quát ta sẽ gọi trục cô lập có véc tơ chỉ phương là \vec{V} và đi qua tâm C_0 của hộp bao thứ nhất, và khi đó phương trình đường thẳng d được xác định như sau: $d = \vec{C}_0 + t \cdot \vec{V}$ với, t là tham số thời gian. \vec{V} là \vec{A}_i, \vec{B}_j hoặc $\vec{A}_i \otimes \vec{B}_j$ với $i, j = 0, 1, 2$.

Khi đó, với điểm P bất kì, hình chiếu của P lên đường thẳng d có gốc C_0 sẽ là điểm H (Hình 2.3). và đoạn thẳng C_0H xác định theo (2.4).

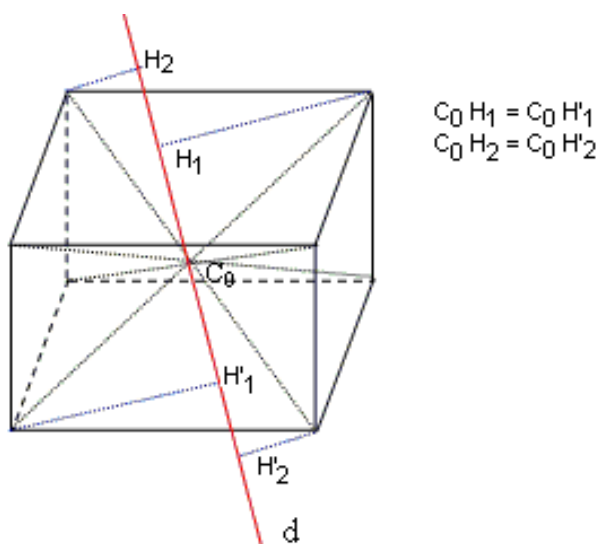


Hình 2.3. Hình chiếu của P lên đường thẳng d

$$hc(P, d) = \frac{(\vec{P} - \vec{C}_0) * \vec{V}}{|\vec{V}|} \quad (2.4)$$

Như vậy, khi chiếu 8 đỉnh của hộp bao thứ nhất lên trục cô lập d với gốc C_0 thì sẽ thu được 4 cặp đoạn thẳng có độ dài bằng nhau nằm về hai phía so với C_0 (Hình 2.4), độ dài của mỗi đoạn thẳng được xác định như sau:

$$hc(\vec{C}_0 + \sum_{i=0}^2 s_i * a_i * \vec{A}_i, d) = \left| \frac{(\sum_{i=0}^2 s_i * a_i * \vec{A}_i) * \vec{V}}{|\vec{V}|} \right| \quad (2.5)$$



Hình 2.4. Chiếu 8 đỉnh của hình hộp lên trục cô lập d

Và khoảng cách nhỏ nhất chứa 8 đoạn thẳng (2.5) với tâm là C_0 và bán kính r_0 được xác định là:

$$r_0 = \max \left\{ \left| \frac{(\sum_{i=0}^2 s_i * a_i * \vec{A}_i) * \vec{V}}{|\vec{V}|} \right| \right\} \text{ Với mọi } |s_i| = 1 \quad (2.6)$$

Đặt $R_0 = r_0 * |\vec{V}|$, ta có:

$$\begin{aligned} \rightarrow R_0 = \max \{ & | a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V} | \\ & | a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V} | \\ & | a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V} | \end{aligned}$$

$$\begin{aligned}
& | a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V} | \\
& |-a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V} | \\
& |-a_0 * \vec{A}_0 * \vec{V} + a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V} | \\
& |-a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} + a_2 * \vec{A}_2 * \vec{V} | \\
& |-a_0 * \vec{A}_0 * \vec{V} - a_1 * \vec{A}_1 * \vec{V} - a_2 * \vec{A}_2 * \vec{V} | \\
& = a_0 * |\vec{A}_0 * \vec{V}| + a_1 * |\vec{A}_1 * \vec{V}| + a_2 * |\vec{A}_2 * \vec{V}|
\end{aligned}$$

Tiếp tục thực hiện tương tự, ta xác định hình chiếu 8 đỉnh của hộp bao thứ hai lên d với góc C_0 như sau.

$$hc(\vec{C}_1 + \sum_{i=0}^2 s_i * b_i * \vec{B}_i, d) = \frac{\vec{V} * \vec{D}}{|\vec{V}|} + \left| \frac{(\sum_{i=0}^2 s_i * b_i * \vec{B}_i) * \vec{V}}{|\vec{V}|} \right| \quad (2.7)$$

$$\text{Với } \vec{D} = \vec{C}_1 - \vec{C}_0$$

Và 8 đoạn thẳng này được nhóm thành 4 cặp đối xứng nhau qua C_1 , và khoảng cách nhỏ nhất chứa 8 đoạn thẳng (2.7) sẽ có tâm là C_1 và bán kính R_1 được xác định:

$$r_1 = \max \left\{ \left| \frac{(\sum_{i=0}^2 s_i * b_i * \vec{B}_i) * \vec{V}}{|\vec{V}|} \right| \right\} \text{ Với mọi } |s_i| = 1 \quad (2.8)$$

Đặt $R_1 = r_1 * |\vec{V}|$, tương tự như trên ta có:

$$R_1 = b_0 * |\vec{B}_0 * \vec{V}| + b_1 * |\vec{B}_1 * \vec{V}| + b_2 * |\vec{B}_2 * \vec{V}|.$$

Hai khoảng cách trên sẽ không giao nhau nếu:

$$C_0 C_1 > r_0 + r_1 \Leftrightarrow |\vec{V}| * C_0 C_1 > |\vec{V}| * r_0 + |\vec{V}| * r_1 \Leftrightarrow R > R_0 + R_1$$

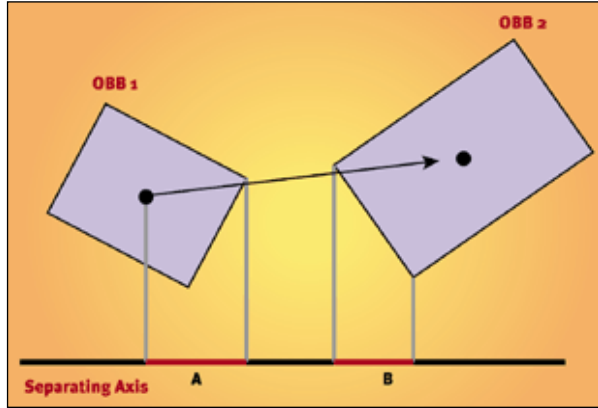
Trong đó: $R = C_0 C_1 * |\vec{V}|$.

Giải cụ thể các phương trình trên. Với mỗi véc tơ ta có:

$$\vec{B}_i = c_{0i} \vec{A}_0 + c_{1i} \vec{A}_1 + c_{2i} \vec{A}_2 \text{ với } i = 0, 1, 2. \quad (2.9)$$

Đặt: $A = (A_0, A_1, A_2)$ và $B = (B_0, B_1, B_2)$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix} \quad (2.10)$$



Hình 2.5. Kết quả chiếu 2 hình hộp lên trục cô lập d [24]

Từ (2.9), (2.10) ta có: $B = C * A \leftrightarrow A^T * B = C$

$$C = \begin{bmatrix} A_0 \\ A_1 \\ A_2 \end{bmatrix} * \begin{bmatrix} B_0 & B_1 & B_2 \end{bmatrix} = \begin{bmatrix} A_0 B_0 & A_0 B_1 & A_0 B_2 \\ A_1 B_0 & A_1 B_1 & A_1 B_2 \\ A_2 B_0 & A_2 B_1 & A_2 B_2 \end{bmatrix} \quad (2.11)$$

Từ (2.10) và (2.11) suy ra: $c_{ij} = \vec{A}_i * \vec{B}_j$, hay c_{ij} chính là tích vô hướng của hai véc tơ A_i, B_j . Mặt khác, từ $B = C * A \Rightarrow A = C^T * B \Rightarrow \vec{A}_i = c_{i0} \vec{B}_0 + c_{i1} \vec{B}_1 + c_{i2} \vec{B}_2$

Tập các trục cô lập ứng viên $\vec{V} = \{ \vec{A}_i, \vec{B}_j, \vec{A}_i \otimes \vec{B}_j \}$ với $i, j = 0, 1, 2$ được dùng để tính toán các hệ số R_0, R_1, R ở trên.

Cụ thể hóa, ta tính cho hai trường hợp $\vec{V} = \vec{A}_i$ và $\vec{V} = \vec{A}_i \otimes \vec{B}_j$:

Xét trường hợp $\vec{V} = \vec{A}_0$:

$$R_0 = a_0 * | \vec{A}_0 * \vec{V} | + a_1 * | \vec{A}_1 * \vec{V} | + a_2 * | \vec{A}_2 * \vec{V} | = a_0.$$

$$R_1 = b_0 * | \vec{B}_0 * \vec{V} | + b_1 * | \vec{B}_1 * \vec{V} | + b_2 * | \vec{B}_2 * \vec{V} | = b_0 * |c_{00}| + b_1 * |c_{01}| + b_2 * |c_{02}|.$$

$$R = \vec{A}_0 * \vec{D}$$

Xét trường hợp $\vec{V} = \vec{A}_0 \otimes \vec{B}_0$:

$$R_0 = a_0 * | \vec{A}_0 * \vec{A}_0 \otimes \vec{B}_0 | + a_1 * | \vec{A}_1 * \vec{A}_0 \otimes \vec{B}_0 | + a_2 * | \vec{A}_2 * \vec{A}_0 \otimes \vec{B}_0 |$$

Mặt khác: $\vec{V} = \vec{A}_0 \otimes \vec{B}_0 = \vec{A}_0 \otimes (c_{00} \vec{A}_0 + c_{10} \vec{A}_1 + c_{20} \vec{A}_2) = c_{10} \vec{A}_2 - c_{20} \vec{A}_1.$

Do vậy: $R_0 = a_1 * |c_{20}| + a_2 * |c_{10}|$

$$R_1 = b_0 * |\vec{B}_0 * \vec{V}| + b_1 * |\vec{B}_1 * \vec{V}| + b_2 * |\vec{B}_2 * \vec{V}|.$$

$$R = \vec{V} * \vec{D} = (c_{10} \vec{A}_2 - c_{20} \vec{A}_1) * \vec{D}$$

Ta viết lại \vec{V} dưới dạng:

$$\vec{V} = \vec{A}_0 \otimes \vec{B}_0 = (c_{00} \vec{B}_0 + c_{01} \vec{B}_1 + c_{02} \vec{B}_2) \otimes \vec{B}_0 = -c_{01} \vec{B}_2 + c_{02} \vec{B}_1$$

$$\rightarrow \vec{B}_0 * \vec{V} = \vec{B}_0 * (-c_{01} \vec{B}_2 + c_{02} \vec{B}_1) = 0$$

$$\vec{B}_1 * \vec{V} = \vec{B}_1 * (-c_{01} \vec{B}_2 + c_{02} \vec{B}_1) = c_{02}$$

$$\vec{B}_2 * \vec{V} = \vec{B}_2 * (-c_{01} \vec{B}_2 + c_{02} \vec{B}_1) = -c_{01}$$

$$\rightarrow R_1 = b_1 * |c_{02}| + b_2 * |c_{01}|$$

Tính toán tương tự với các trường hợp còn lại, ta có bảng các giá trị tính toán sẵn cho R, R_0, R_1 (Bảng 2.1).

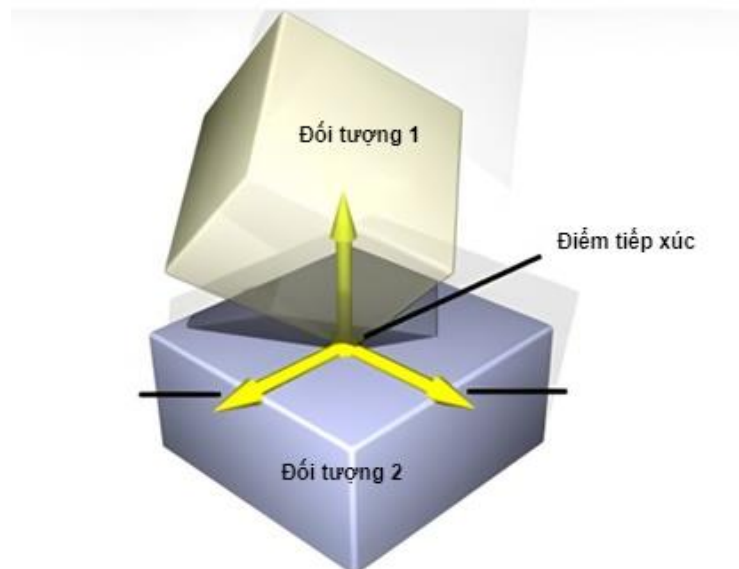
Bảng 2.1. Bảng các giá trị R, R_0, R_1 được tính toán trước.

\vec{V}	R_0	R_1	R
\vec{A}_0	a_0	$b_0 c_{00} +b_1 c_{01} +b_2 c_{02} $	$ \vec{A}_0 \cdot \vec{D} $
\vec{A}_1	a_1	$b_0 c_{10} +b_1 c_{11} +b_2 c_{12} $	$ \vec{A}_1 \cdot \vec{D} $
\vec{A}_2	a_2	$b_0 c_{20} +b_1 c_{21} +b_2 c_{22} $	$ \vec{A}_2 \cdot \vec{D} $
\vec{B}_0	$a_0 c_{00} +a_1 c_{10} +a_2 c_{20} $	b_0	$ \vec{B}_0 \cdot \vec{D} $
\vec{B}_1	$a_0 c_{01} +a_1 c_{11} +a_2 c_{21} $	b_1	$ \vec{B}_1 \cdot \vec{D} $
\vec{B}_2	$a_0 c_{02} +a_1 c_{12} +a_2 c_{22} $	b_2	$ \vec{B}_2 \cdot \vec{D} $
$\vec{A}_0 \times \vec{B}_0$	$a_1 c_{20} +a_2 c_{10} $	$b_1 c_{02} +b_2 c_{01} $	$c_{10} \cdot \vec{A}_2 \cdot \vec{D} - c_{20} \cdot \vec{A}_1 \cdot \vec{D}$
$\vec{A}_0 \times \vec{B}_1$	$a_1 c_{21} +a_2 c_{11} $	$b_0 c_{02} +b_2 c_{00} $	$c_{11} \cdot \vec{A}_2 \cdot \vec{D} - c_{21} \cdot \vec{A}_1 \cdot \vec{D}$
$\vec{A}_0 \times \vec{B}_2$	$a_1 c_{22} +a_2 c_{12} $	$b_0 c_{01} +b_1 c_{00} $	$c_{12} \cdot \vec{A}_2 \cdot \vec{D} - c_{22} \cdot \vec{A}_1 \cdot \vec{D}$
$\vec{A}_1 \times \vec{B}_0$	$a_0 c_{20} +a_2 c_{00} $	$b_1 c_{12} +b_2 c_{11} $	$c_{20} \cdot \vec{A}_0 \cdot \vec{D} - c_{00} \cdot \vec{A}_2 \cdot \vec{D}$
$\vec{A}_1 \times \vec{B}_1$	$a_0 c_{21} +a_2 c_{01} $	$b_0 c_{12} +b_2 c_{10} $	$c_{21} \cdot \vec{A}_0 \cdot \vec{D} - c_{01} \cdot \vec{A}_2 \cdot \vec{D}$
$\vec{A}_1 \times \vec{B}_2$	$a_0 c_{22} +a_2 c_{02} $	$b_0 c_{11} +b_1 c_{10} $	$c_{22} \cdot \vec{A}_0 \cdot \vec{D} - c_{02} \cdot \vec{A}_2 \cdot \vec{D}$
$\vec{A}_2 \times \vec{B}_0$	$a_0 c_{10} +a_1 c_{00} $	$b_1 c_{22} +b_2 c_{21} $	$c_{00} \cdot \vec{A}_1 \cdot \vec{D} - c_{10} \cdot \vec{A}_0 \cdot \vec{D}$
$\vec{A}_2 \times \vec{B}_1$	$a_0 c_{11} +a_1 c_{01} $	$b_0 c_{22} +b_2 c_{20} $	$c_{01} \cdot \vec{A}_1 \cdot \vec{D} - c_{11} \cdot \vec{A}_0 \cdot \vec{D}$
$\vec{A}_2 \times \vec{B}_2$	$a_0 c_{12} +a_1 c_{02} $	$b_0 c_{21} +b_1 c_{20} $	$c_{02} \cdot \vec{A}_1 \cdot \vec{D} - c_{12} \cdot \vec{A}_0 \cdot \vec{D}$

2.1.2.2. Tìm điểm va chạm

Việc phát hiện nhanh va chạm ở mức một sẽ chỉ xác định được việc có hay không việc xảy ra va chạm giữa hai hộp bao theo hướng của các đối tượng, do vậy ứng dụng chỉ dừng lại mức cung cấp thông tin và cảnh báo va chạm. Để giải quyết bài toán tổng quát, phần này sẽ nghiên cứu phương pháp tính toán chính xác điểm va chạm của hai hộp bao của các đối tượng.

Nhận xét rằng, thời điểm đầu tiên khi hai đối tượng xảy ra va chạm chính là thời điểm chúng có tiếp xúc nhau, nói cách khác là thời điểm chúng chạm vào nhau và gây ra hậu quả va chạm. Vậy nên, để xác định điểm va chạm ta sẽ quan tâm đến thời điểm và điểm mà các đối tượng tiếp xúc nhau (Hình 2.6).



Hình 2.6. Tìm điểm va chạm khi hai đối tượng tiếp xúc nhau [24]

Đối với hai hộp bao, nếu chúng va chạm với nhau ở dạng đỉnh - đỉnh, đỉnh - cạnh, cạnh - cạnh, đỉnh - mặt thì điểm tiếp xúc là duy nhất. Nhưng nếu chúng va chạm với nhau ở dạng mặt - mặt, cạnh - mặt thì sẽ có vô số điểm tiếp xúc, khi đó ta chỉ cần đưa ra một điểm bất kì là được.

Ý tưởng tìm thời điểm va chạm: mỗi khi thực hiện công việc kiểm tra nhanh va chạm ở mức một, nếu tìm được một trục cô lập thì ta sẽ ghi lại nhãn thời gian cho trục cô lập đó. Nếu không tìm được một trục cô lập nào, nghĩa là hai hộp bao đã va chạm với nhau, khi đó nhãn thời gian được gán cho trục cô lập ở lần kiểm tra trước đó sẽ là thời điểm đầu tiên mà hai hộp bao va chạm nhau. Gọi T là nhãn thời gian đó, khi đó ta có thể coi như $R = R_0 + R_1$ (thời điểm hai hình hộp tiếp xúc nhau).

Gọi P là điểm tiếp xúc của hai hộp bao tại thời điểm T , ta sẽ phải tìm P thỏa mãn công thức 2.12

$$\sum_{i=0}^2 x_i * \vec{A}_i = \vec{D} + \sum_{j=0}^2 y_j * \vec{B}_j \quad (2.12)$$

Với $|x_i| \leq a_i, |y_j| \leq b_j$ với $i, j = 0, 1, 2$.

Ở đây: $\vec{D} = (C_1 - T\vec{v}_1) - (C_0 - T\vec{v}_0)$ với \vec{v}_0, \vec{v}_1 : vận tốc tương ứng của hai hộp bao; C_0, C_1 : tâm tương ứng của hai hộp bao.

Việc tìm điểm va chạm sẽ phụ thuộc vào trục cô lập \vec{V} ở thời điểm T là trục nào trong số tập ứng viên được xác định (15 trục cô lập).

Xét 3 trường hợp:

1- \vec{V} là véc tơ \vec{A}_i :

Nhân hai vế của (2.12) với \vec{A}_i ta thu được: $x_i = \vec{A}_i * \vec{D} + \sum_{j=0}^2 y_j * \vec{A}_i \vec{B}_j$

Nếu $\vec{A}_i * \vec{D} > 0$ thì $\vec{A}_i * \vec{D} = R_0 + R_1$ từ hai khoảng cắt nhau tại điểm mút bên phải của $[-R_0, R_0]$.

Nếu $\vec{A}_i * \vec{D} < 0$ thì $\vec{A}_i * \vec{D} = R_0 + R_1$ từ hai khoảng cắt nhau tại điểm mút bên trái của $[-R_0, R_0]$.

Do đó, $\vec{A}_i * \vec{D} = \sigma(R_0 + R_1)$ với $|\sigma| = 1$, khi đó:

$$x_i = \sigma * (R_0 + R_1) + \sum_{j=0}^2 y_j * c_{ij}$$

Thay các giá trị của R_0, R_1 trong bảng 1.1 ta có:

$$\begin{aligned}
x_i &= \sigma * \left(a_i + \sum_{j=0}^2 b_j * |c_{ij}| \right) + \sum_{j=0}^2 y_j * c_{ij} \\
&= \sigma * \left(a_i + \sum_{j=0}^2 b_j * |c_{ij}| \right) + \sum_{j=0}^2 y_j * c_{ij} \\
0 &= \sum_{j=0}^2 |c_{ij}| * (b_j + \sigma * \text{sign}(c_{ij}) * y_j) + (a_i - \sigma * x_i)
\end{aligned} \tag{2.13}$$

Ở đây, $(a_i - \sigma * x_i) \geq 0$ và $(b_j + \sigma * \text{sign}(c_{ij}) * y_j) \geq 0$

$$\rightarrow \begin{cases} (a_i - \sigma * x_i) = 0 \\ (b_j + \sigma * \text{sign}(c_{ij}) * y_j) = 0 \end{cases}$$

$$\text{Nếu } c_{ij} \neq 0 \rightarrow \begin{cases} x_i = \sigma * a_i \\ y_j = -\sigma * \text{Sign}(c_{ij}) * b_j \quad j = 0, 1, 2. \end{cases}$$

Nếu $c_{ij} = 0$ (ứng với trường hợp *cạnh va chạm mặt, mặt va chạm mặt*). Khi đó, nhân hai vế của (2.12) với \vec{B}_j ta được:

$$\begin{aligned}
y_j &= -\vec{B}_j * \vec{D} + \sum_{k=0}^2 x_k * c_{kj} \\
\rightarrow \min(y_j) &= -\vec{B}_j * \vec{D} - \sum_{k=0}^2 |c_{kj}| a_k \leq y_j \leq -\vec{B}_j * \vec{D} + \sum_{k=0}^2 |c_{kj}| a_k = \max(y_j)
\end{aligned}$$

Vì $|y_j| \leq b_j$ nên $y_j \in [\min(y_j), \max(y_j)] \cap [-b_j, b_j]$. Và do đó, ta chỉ chọn một giá trị y_j thuộc đoạn trên.

2- \vec{V} là véc tơ \vec{B}_i : Tương tự ta có:

$$\text{Nếu } c_{ij} \neq 0 \rightarrow \begin{cases} y_i = -\sigma * b_i \\ x_j = -\sigma * \text{Sign}(c_{ij}) * a_j \quad j = 0, 1, 2. \end{cases}$$

Nếu $c_{ij} = 0$, nhân hai vế của (2.12) với \vec{A}_i ta có:

$$\begin{aligned}
x_j &= \vec{A}_j * \vec{D} + \sum_{k=0}^2 y_k c_{jk} \\
\rightarrow \min(x_j) &= -\vec{A}_j * \vec{D} - \sum_{k=0}^2 |c_{jk}| b_k \leq x_j \leq -\vec{A}_j * \vec{D} + \sum_{k=0}^2 |c_{jk}| b_k = \max(x_j)
\end{aligned}$$

Việc chọn điểm tiếp xúc x_j thuộc đoạn:

$$x_j \in [\min(x_j), \max(x_j)] \cap [-a_j, a_j]$$

3- \vec{V} là véc tơ $\vec{A}_i \otimes \vec{B}_j$: Ta tính toán cho trường hợp $\vec{V} = \vec{A}_0 \otimes \vec{B}_0 = c_{10}\vec{A}_2 - c_{20}\vec{A}_1 = -c_{01}\vec{B}_2 + c_{02}\vec{B}_1$, và các trường hợp khác sẽ được tính tương tự. Cụ thể:

Nhân hai vế (2.12) với $\vec{A}_0 \otimes \vec{B}_0$ ta được:

$$\begin{aligned} x_2 * c_{10} - x_1 * c_{20} &= (\vec{A}_0 \otimes \vec{B}_0) * \vec{D} = (\vec{A}_0 \otimes \vec{B}_0) * \vec{D} + \sum_{j=0}^2 y_j * \vec{B}_j * (-c_{01}\vec{B}_2 + c_{02}\vec{B}_1) \\ x_2 * c_{10} - x_1 * c_{20} &= (\vec{A}_0 \otimes \vec{B}_0) * \vec{D} = (\vec{A}_0 \otimes \vec{B}_0) * \vec{D} + y_1 * c_{02} - y_2 * c_{01} \end{aligned} \quad (2.14)$$

Theo hàng 7 trong bảng 1.1:

$$\begin{aligned} |R| &= |(\vec{A}_0 \otimes \vec{B}_0) * \vec{D}| = R_0 + R_1 = a_1 * |c_{20}| + a_2 * |c_{10}| + b_1 * |c_{02}| + b_2 * |c_{01}| \\ \rightarrow R &= \text{Sign}((\vec{A}_0 \otimes \vec{B}_0) * \vec{D}) * \{ a_1 * |c_{20}| + a_2 * |c_{10}| + b_1 * |c_{02}| + b_2 * |c_{01}| \} \end{aligned}$$

Đặt $\sigma = \text{Sign}((\vec{A}_0 \otimes \vec{B}_0) * \vec{D})$, ta có (2.12) tương đương với:

$$\begin{aligned} x_2 * c_{10} - x_1 * c_{20} &= \sigma * (a_1 * |c_{20}| + a_2 * |c_{10}| + b_1 * |c_{02}| + b_2 * |c_{01}|) + y_1 * c_{02} - y_2 * c_{01} \\ &\Rightarrow |c_{20}| * (a_1 + s * \text{Sign}(c_{20}) * x_1) \\ &\quad + |c_{10}| * (a_2 - s * \text{Sign}(c_{10}) * x_2) \\ &\quad + |c_{01}| * (b_2 - s * \text{Sign}(c_{01}) * y_2) \\ &\quad + |c_{02}| * (b_1 + s * \text{Sign}(c_{02}) * y_1) = 0 \end{aligned} \quad (2.15)$$

Có:

$$\begin{cases} a_1 + s * \text{Sign}(c_{20}) * x_1 \geq 0 \\ a_2 - s * \text{Sign}(c_{10}) * x_2 \geq 0 \\ b_2 - s * \text{Sign}(c_{01}) * y_2 \geq 0 \\ b_1 + s * \text{Sign}(c_{02}) * y_1 \geq 0 \end{cases} \quad \text{Từ (2.15)} \Rightarrow \begin{cases} a_1 + s * \text{Sign}(c_{20}) * x_1 = 0 \\ a_2 - s * \text{Sign}(c_{10}) * x_2 = 0 \\ b_2 - s * \text{Sign}(c_{01}) * y_2 = 0 \\ b_1 + s * \text{Sign}(c_{02}) * y_1 = 0 \end{cases}$$

$$\Rightarrow \begin{cases} x_1 = -\sigma * \text{Sign}(c_{20}) * a_1 \\ x_2 = \sigma * \text{Sign}(c_{10}) * a_2 \\ y_1 = -\sigma * \text{Sign}(c_{02}) * b_1 \\ y_2 = \sigma * \text{Sign}(c_{01}) * b_2 \end{cases}$$

Để tìm x_0 và y_0 , nhân 2 vế của (2.12) với \vec{A}_0 và \vec{B}_0 :

$$\begin{cases} x_0 = \vec{A}_0 * \vec{D} + c_{00} * y_0 + c_{01} * y_1 + c_{02} * y_2 \\ c_{00} * x_0 + c_{10} * x_1 + c_{20} * x_2 = \vec{B}_0 * \vec{D} + y_0 \end{cases}$$

Giải hệ phương trình trên, cùng với các hệ số y_1, y_2, x_1, x_2 ở trên:

$$\begin{cases} x_0 = \frac{1}{1-c_{00}^2} \left[\vec{A}_0 * \vec{D} + c_{00} * (-\vec{B}_0 * \vec{D} + c_{10} * x_1 + c_{20} * x_2) + c_{01} * y_1 + c_{02} * y_2 \right] \\ y_0 = \frac{1}{1-c_{00}^2} \left[-\vec{B}_0 * \vec{D} + c_{00} * (\vec{A}_0 * \vec{D} + c_{01} * y_1 + c_{02} * y_2) + c_{10} * x_1 + c_{20} * x_2 \right] \end{cases}$$

Đối với các trường hợp $\vec{V} = \{ \vec{A}_0 \otimes \vec{B}_1, \vec{A}_0 \otimes \vec{B}_2, \vec{A}_1 \otimes \vec{B}_0, \vec{A}_1 \otimes \vec{B}_1, \vec{A}_1 \otimes \vec{B}_2, \vec{A}_2 \otimes \vec{B}_0, \vec{A}_2 \otimes \vec{B}_1, \vec{A}_2 \otimes \vec{B}_2 \}$, ta tính toán tương tự để tìm điểm tiếp xúc.

Tổng hợp các kết quả tính toán, ta xây dựng được bảng tra cứu tọa độ các tiếp điểm (Bảng 2.2):

Bảng 2.2. Bảng tính toán tọa độ của tiếp điểm trong mọi trường hợp

\vec{V}	Tiếp điểm
\vec{A}_i	$y_j = -\sigma * \text{Sign}(c_{ij}) * b_j \quad j = 0, 1, 2.$
\vec{B}_j	$x_j = -\sigma * \text{Sign}(c_{ij}) * a_j \quad j = 0, 1, 2.$
$\vec{A}_0 \times \vec{B}_0$	$x_1 = -\sigma * \text{Sign}(c_{20}) * a_1; x_2 = \sigma * \text{Sign}(c_{10}) * a_2;$ $y_1 = -\sigma * \text{Sign}(c_{02}) * b_1; y_2 = \sigma * \text{Sign}(c_{01}) * b_2$ $x_0 = \frac{1}{1-c_{00}^2} \left[\vec{A}_0 * \vec{D} + c_{00} * (-\vec{B}_0 * \vec{D} + c_{10} * x_1 + c_{20} * x_2) + c_{01} * y_1 + c_{02} * y_2 \right]$
$\vec{A}_0 \times \vec{B}_1$	$x_1 = -\sigma * \text{Sign}(c_{21}) * a_1; x_2 = \sigma * \text{Sign}(c_{11}) * a_2;$ $y_1 = -\sigma * \text{Sign}(c_{02}) * b_0; y_2 = \sigma * \text{Sign}(c_{00}) * b_2$ $x_0 = \frac{1}{1-c_{01}^2} \left[\vec{A}_0 * \vec{D} + c_{01} * (-\vec{B}_1 * \vec{D} + c_{11} * x_1 + c_{21} * x_2) + c_{00} * y_0 + c_{02} * y_2 \right]$

$\vec{A}_0 \times \vec{B}_2$	$x_1 = -\sigma * \text{Sign}(c_{22}) * a_1; x_2 = \sigma * \text{Sign}(c_{12}) * a_2;$ $y_1 = -\sigma * \text{Sign}(c_{01}) * b_0; y_2 = \sigma * \text{Sign}(c_{00}) * b_1$ $x_0 = \frac{1}{1-c_{02}^2} \left[\vec{A}_0 * \vec{D} + c_{02} * (-\vec{B}_2 * \vec{D} + c_{12} * x_1 + c_{22} * x_2) + c_{00} * y_1 + c_{01} * y_2 \right]$
$\vec{A}_1 \times \vec{B}_0$	$x_0 = \sigma * \text{Sign}(c_{20}) * a_0; x_2 = -\sigma * \text{Sign}(c_{00}) * a_2;$ $y_1 = -\sigma * \text{Sign}(c_{12}) * b_1; y_2 = \sigma * \text{Sign}(c_{11}) * b_2$ $x_1 = \frac{1}{1-c_{10}^2} \left[\vec{A}_1 * \vec{D} + c_{10} * (-\vec{B}_0 * \vec{D} + c_{00} * x_0 + c_{20} * x_2) + c_{11} * y_1 + c_{12} * y_2 \right]$
$\vec{A}_1 \times \vec{B}_1$	$x_0 = \sigma * \text{Sign}(c_{21}) * a_0; x_2 = -\sigma * \text{Sign}(c_{01}) * a_2;$ $y_1 = \sigma * \text{Sign}(c_{10}) * b_1; y_2 = -\sigma * \text{Sign}(c_{10}) * b_2$ $x_1 = \frac{1}{1-c_{11}^2} \left[\vec{A}_1 * \vec{D} + c_{11} * (-\vec{B}_1 * \vec{D} + c_{01} * x_0 + c_{21} * x_2) + c_{10} * y_1 + c_{12} * y_2 \right]$
$\vec{A}_1 \times \vec{B}_2$	$x_0 = \sigma * \text{Sign}(c_{22}) * a_0; x_2 = -\sigma * \text{Sign}(c_{02}) * a_2;$ $y_1 = -\sigma * \text{Sign}(c_{11}) * b_0; y_2 = \sigma * \text{Sign}(c_{10}) * b_1$ $x_1 = \frac{1}{1-c_{12}^2} \left[\vec{A}_1 * \vec{D} + c_{12} * (-\vec{B}_2 * \vec{D} + c_{02} * x_0 + c_{22} * x_2) + c_{10} * y_1 + c_{11} * y_2 \right]$
$\vec{A}_2 \times \vec{B}_0$	$x_0 = -\sigma * \text{Sign}(c_{10}) * a_0; x_2 = \sigma * \text{Sign}(c_{00}) * a_1;$ $y_1 = -\sigma * \text{Sign}(c_{22}) * b_1; y_2 = \sigma * \text{Sign}(c_{21}) * b_1$ $x_2 = \frac{1}{1-c_{20}^2} \left[\vec{A}_2 * \vec{D} + c_{20} * (-\vec{B}_0 * \vec{D} + c_{00} * x_0 + c_{10} * x_1) + c_{21} * y_1 + c_{22} * y_2 \right]$
$\vec{A}_2 \times \vec{B}_1$	$x_0 = -\sigma * \text{Sign}(c_{11}) * a_0; x_2 = \sigma * \text{Sign}(c_{01}) * a_1;$ $y_0 = \sigma * \text{Sign}(c_{22}) * b_0; y_2 = -\sigma * \text{Sign}(c_{20}) * b_2$ $x_2 = \frac{1}{1-c_{21}^2} \left[\vec{A}_2 * \vec{D} + c_{21} * (-\vec{B}_1 * \vec{D} + c_{01} * x_0 + c_{11} * x_1) + c_{20} * y_0 + c_{22} * y_2 \right]$
$\vec{A}_2 \times \vec{B}_2$	$x_0 = -\sigma * \text{Sign}(c_{12}) * a_0; x_2 = \sigma * \text{Sign}(c_{02}) * a_1;$ $y_0 = -\sigma * \text{Sign}(c_{21}) * b_0; y_1 = \sigma * \text{Sign}(c_{20}) * b_1$ $x_2 = \frac{1}{1-c_{22}^2} \left[\vec{A}_2 * \vec{D} + c_{22} * (-\vec{B}_2 * \vec{D} + c_{02} * x_0 + c_{12} * x_1) + c_{20} * y_0 + c_{21} * y_2 \right]$

Như vậy, với kĩ thuật tính toán va chạm dựa trên hộp bao theo hướng, cả hai mức phát hiện va chạm của thuật toán đã tạo lập cho ta bảng dữ liệu chứa thông tin kết quả của việc xác định có va chạm hay không (Bảng 2.1) và điểm va chạm của hai hộp bao nếu có va chạm (Bảng 2.2). Do vậy, quá trình phát hiện va chạm được thực

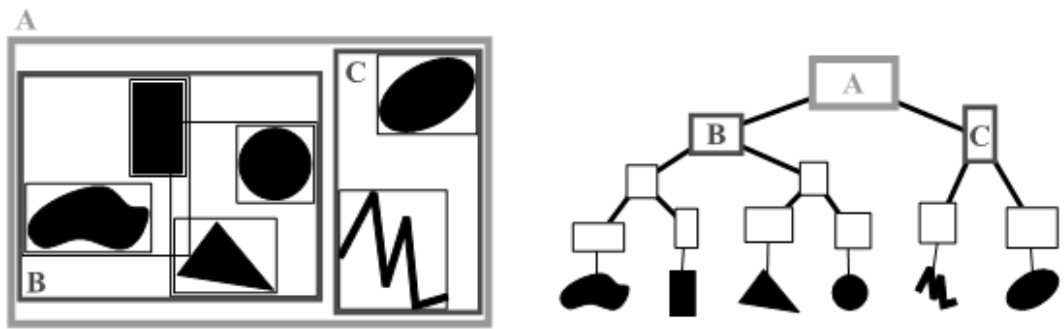
hiện nhanh chóng và kết quả thực nghiệm cho thấy việc phân chia các mức phát hiện va chạm được cải thiện đáng kể về hiệu quả xử lý và tính toán. Ứng dụng đạt được cả với mô hình có nhiều đối tượng cùng đồng thời tham gia, khả năng xảy ra va chạm và số lần xảy ra va chạm ít, rất thưa.

2.1.3. Phát hiện, tính toán va chạm sử dụng hệ bao

Hệ bao (bounding volume hierarchy) là một cấu trúc cây đối với một nhóm các đối tượng hình học được tổ chức, tạo bởi các hình bao (bounding volume). Trong đó, hệ bao là cây bao trùm với các hình bào là các node của cây. Các node được gom thành các nhóm nhỏ và tiếp tục được đóng kín trong các hình bao lớn hơn. Quá trình tạo cây được lặp lại và cuối cùng tạo nên một cấu trúc cây với một hình bao đơn tại đỉnh của cây.

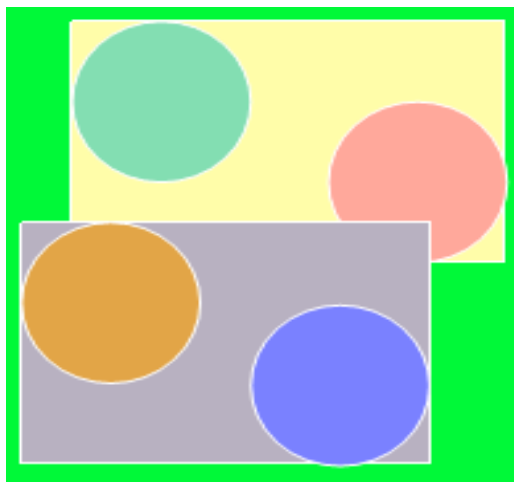
Hiện nay, hệ bao được ứng dụng trong đồ họa máy tính, phát hiện va chạm và thuật toán dò tia nhằm mục đích tăng tốc độ tính toán. Đặc biệt, trong bài toán phát hiện, tránh va chạm thì hệ bao đang thể hiện rõ ưu điểm bằng việc giảm số lần kiểm tra va chạm theo cặp, cải thiện hiệu suất tổng thể [84, 86].

Việc xây dựng hệ bao được nhiều nhóm tác giả nghiên cứu và phát triển dựa trên một số kỹ thuật bao gồm chia cắt ở giữa các trục dài nhất, chia cắt ở giá trị trung bình, chia cắt dựa trên thuật toán SAH (Surface Area Heuristic). Trong số này, phương pháp chia cắt ở giá trị trung bình (Median Cut) thường được lựa chọn bởi tính đơn giản trong cài đặt, và hiệu quả trong việc thực hiện tạo lập hệ bao. Ý tưởng của phương pháp này là chia cắt node gốc thành các node (trái/phải) dựa trên giá trị trung vị của khối bao. Bắt đầu từ một hộp bao quanh toàn bộ đối tượng, thuật toán tìm giá trị trung vị của một trục và chia các điểm thành hai nhóm theo giá trị này, sau cùng là chia thành hai hộp bao với các giá trị điểm trung vị xác định. Quá trình này tiếp tục được tính cho từng nhánh của cây gốc cho đến khi số lượng điểm trong mỗi nhánh đạt mức nhất định. Kết thúc, ta thu được một hệ bao với gốc là hộp bao lớn nhất của các đối tượng, và các hệ bao con, hộp bao bao lấy nhóm, một nhóm hay một đối tượng.

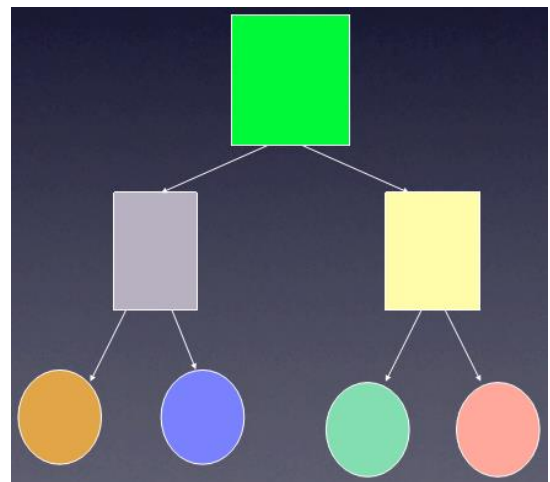


Hình 2.7. Hệ bao BVH sử dụng phương pháp khối bao chữ nhật.

Thuật toán được mô tả như sau (Hình 2.7 và Hình 2.8): các đối tượng được sắp xếp dọc theo một trong ba trục tọa độ x,y,z; xác định giá trị trung bình tọa độ của các đối tượng trên trục tọa độ đó; tiến hành cắt đối tượng đó thành hai node tương ứng là node trái và node phải.



a. Node gốc của đối tượng trước khi xây dựng cây phân lớp



b. Cây phân lớp sau khi được xây dựng.

Hình 2.8. Xây dựng hệ bao BVH sử dụng phương pháp Median-Cut

Các bước thực hiện của thuật toán Media Cut

1. Xác định hình bao mọi đối tượng (node gốc).
2. Chọn trục sắp xếp các đối tượng trong node.
3. Cập nhật các giá trị tối thiểu và tối đa của mọi hộp bao của mọi đối tượng trong nút hiện tại theo trục đã chọn.
4. Tính giá trị trung vị

5. Chia node gốc thành hai node con, một chứa các đối tượng nhỏ hơn giá trị trung vị và một có giá trị lớn hơn trung vị
6. Lặp lại quá trình chi tách hộp bao thành phần cho đến mức đạt yêu cầu.

Để phát hiện va chạm của hai cấu trúc B1, B2, ta kiểm tra sự va chạm của hai node gốc, nếu kết quả kiểm tra va chạm ở hai node gốc là không thì sẽ không xảy ra va chạm của hai đối tượng đó, quá trình kiểm tra va chạm kết thúc. Ngược lại, nếu hai node gốc của B1, B2 có va chạm, việc va chạm đó sẽ được xảy ra với một trong những node con bên trái hoặc bên phải của B1 với B2 (hoặc ngược lại). Với trường hợp này, tiến hành tinh chỉnh, cập nhật cấu trúc phân cấp thay thế các node con để xây dựng cấu trúc phân cấp mới. Quá trình đó được đệ quy để tiếp tục kiểm tra va chạm của các cấu trúc hệ bao mới.

Thuật toán phát hiện va chạm của hai hệ bao:

Dữ liệu đầu vào: cấu trúc hệ bao B1, B2.

Kết quả trả về: trạng thái có/không xảy ra va chạm.

Mô tả thuật toán:

Khởi tạo	$B1 \leftarrow \text{root}(OBB1)$ $B2 \leftarrow \text{root}(OBB2)$
Thuật toán phát hiện va chạm của 2 hệ bao B1 và B2	<pre> if B1 does not intersects B2 then RETURN FALSE else if B1 is bigger than B2 then SWAP B1 and B2 {Ensure B2 is bigger than B1} end if {Refine the big bounding volume hierarchy B2} for all child C of B2 do if Detect collision between C and B1 then RETURN TRUE end if end for RETURN FALSE end if </pre>
Kết quả	Trả về trạng thái có/không xảy ra va chạm giữa B1 và B2

Như vậy, thuật toán phát hiện va chạm của hai hệ bao là một dạng duyệt cây nhị phân đối với việc phát hiện va chạm của hai cấu trúc cây B1, B2. Trong trường hợp xấu nhất, giả thiết rằng N1, N2 là chiều cao tương ứng của cấu trúc B1, B2, thì độ phức tạp tìm kiếm trên cây nhị phân trong trường hợp tốt nhất của của hai cấu trúc B1, B2 là $O(\log(N))$, trong trường hợp xấu nhất độ phức tạp là $O(N)$.

Với $N = \max(N1, N2)$

Như vậy, thuật toán bảo đảm hiệu năng tính toán hiện nay và có thể ứng dụng được cho việc phát hiện va chạm thực tế.

2.2. Phương pháp tránh va chạm Elastic strips

2.2.1. Giới thiệu

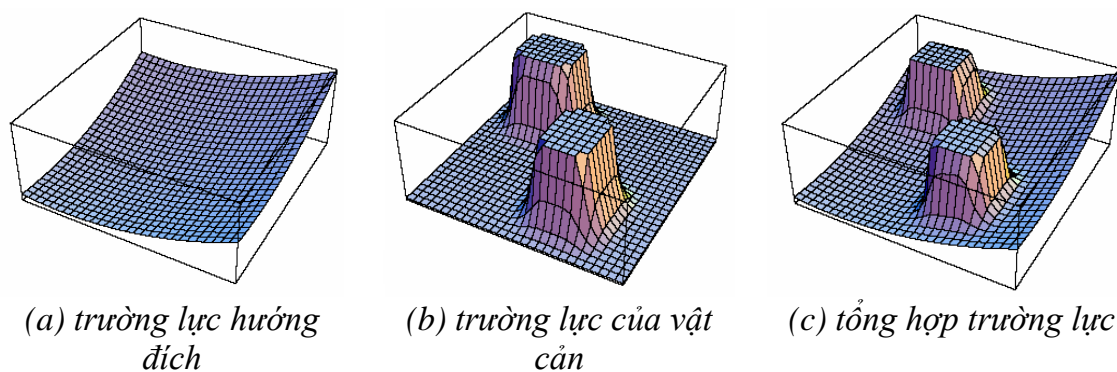
Trong bài toán tránh va chạm, kỹ thuật cơ bản và quan trọng hàng đầu đó là bảo đảm độ an toàn trong quá trình tự vận hành của robot trong không gian làm việc, di chuyển. Các thuật toán, kỹ thuật tránh va chạm này đều dựa vào hệ thống cảm biến (khoảng cách đến các đối tượng, đặc tính vật lý kỹ thuật của môi trường) của robot để đưa ra các hành vi tương ứng giúp robot tránh, vượt các vật cản.

Elastic strips là một phương pháp được nhiều nhà nghiên cứu đánh giá là có hiệu quả trong giải quyết bài toán tránh vật cản cho robot tự hành thông qua việc kiểm thử thuật toán gốc [36, 51]. Kết quả các nghiên cứu cho thấy, thuật toán giúp cho các robot tự hành có thể tránh được va chạm với các vật cản cả trong môi trường cố định và môi trường động. Đồng thời, hoàn toàn phù hợp với hệ thời gian thực thông qua các hiệu quả tính toán, độ chính xác của đường đi, và tốc độ tính toán của hệ thống.

Trong công trình [40, 41], tác giả O. Khatib giả thiết rằng quỹ đạo dưới tác động của các lực hướng đích, phản lực (nếu có) từ các vật cản thì dải đàn hồi sẽ được điều chỉnh và cập nhật tại mỗi thời điểm cụ thể trong quá trình di chuyển của robot. Trong đó: lực hướng tâm (nội lực) (Hình 2.9a) là lực được tạo nên bởi robot và điểm đích đến, lực này được coi như lực kéo phát sinh từ điểm đích và luôn có xu hướng tác động (lực, phương) kéo robot về phía điểm đích; phản lực (ngoại lực) (Hình 2.9b)

là lực được hình thành nếu như trong quá trình dịch chuyển, robot gặp các vật cản và giả thiết rằng khi đó nếu di chuyển tại thời điểm $(t+1)$ thì sẽ xảy ra va chạm, do vậy lực này được coi là giả lực có tác động từ vật cản lên robot nếu như có thể xảy ra va chạm. Tổng hợp của hai véc tơ lực (Hình 2.9c) cho ta một trường lực tổng hợp và dựa trên véc tơ lực tổng hợp này, việc tính toán vị trí tiếp theo cho robot được thực hiện.

Phương pháp Elastic strips cho thấy khả năng dẫn hướng cho robot trong quá trình di chuyển ổn định và hiệu quả hơn so với phương pháp APF đã đề xuất ở trên. Các ứng dụng của Elastic strips cũng đã được cài đặt thuận lợi và dễ dàng cho hệ thống robot nhiều bậc tự do, robot tay máy. Sự kết hợp nhịp nhàng giữa khâu lập lịch di chuyển với khâu thực thi chuyển động của robot đã được kiểm chứng và đem lại kết quả tốt. Đồng thời, thuật toán tìm đường cho robot cũng đã được xây dựng, cài đặt hướng đến việc tối ưu hóa và hỗ trợ cho robot có thể di chuyển theo quỹ đạo đã được thiết kế, có khả năng vượt các vật cản trong quá trình di chuyển và đặc biệt hơn, thuật toán này phù hợp với những môi trường thực tế (trong đó các đối tượng di chuyển theo các quỹ đạo riêng của nó).



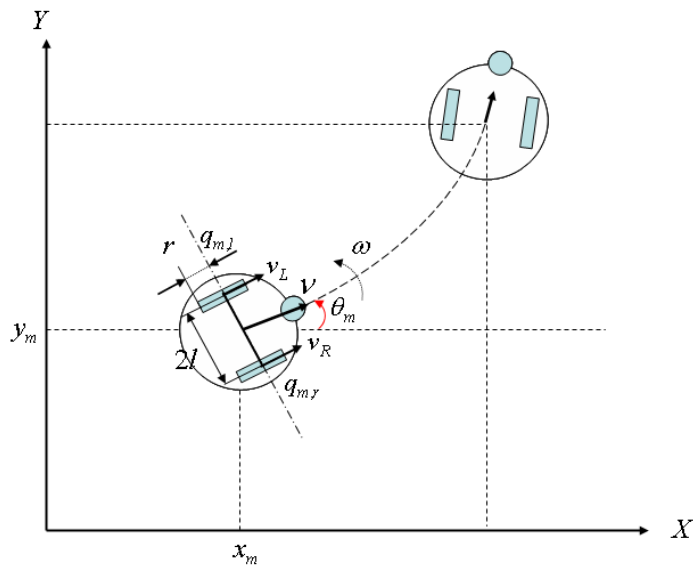
Hình 2.9. Định nghĩa trường lực tiềm năng

Trong hệ thời gian thực, việc kiểm soát và tác động lên hoạt động của robot tại từng thời điểm cố định mang tính chất tuần hoàn luôn là một nhân tố quan trọng trong quá trình hoạt động của robot, các thuật toán được cài đặt, nghiên cứu đều phải đáp ứng đầy đủ, phù hợp với hệ thời gian thực: thuật toán đơn giản, cài đặt thuận tiện, dễ dàng và hoạt động nhanh, hiệu quả. Đáp ứng các yêu cầu đó, phương pháp Elastic

strips được đề xuất đảm bảo tốt các yêu cầu nêu trên và hoàn toàn phù hợp cho robot hoạt động trong hệ thời gian thực. Phương pháp này được cài đặt, kiểm thử và đánh giá là phù hợp với hệ thời gian thực trong môi trường xác định [36, 51].

Trong công trình [19, 42], O. Khatib đã đề xuất một phương pháp tránh va chạm cho robot đó là “*Dải đàn hồi- Elastic strips*”, ý tưởng đề ra đó là việc xác định robot di chuyển từ vị trí xuất phát đến điểm đích trên một quỹ đạo cố định và trong quá trình di chuyển, robot sẽ gặp các vật cản và lúc này quỹ đạo của robot được thay đổi theo lý thuyết của một dải dây đàn hồi. Khi tiếp cận với vật cản, trường lực ảo sẽ được phát sinh giữa robot và vật cản dựa trên lý thuyết về trường lực tiềm năng (*Potential Field*).

Hình 2.10 dưới đây giới thiệu mô hình robot với hệ thống di chuyển và được biểu diễn trên hệ trục Descartes.



Hình 2.10. Mô hình robot và hệ trục tọa độ.

Trong đó:

X, Y : Hệ tọa độ Descartes.

(x_m, y_m) : Vị trí của robot trong hệ tọa độ Descartes.

(q_{ml}, q_{mr}) : Vận tốc góc của bánh xe bên trái và bên phải.

l : Khoảng cách từ tâm của robot đến mỗi bánh xe.

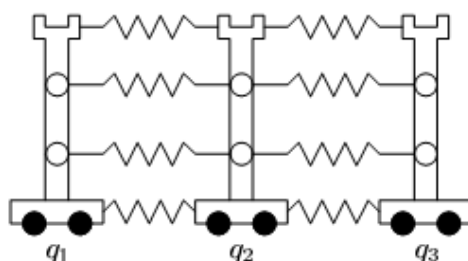
r : Vận tốc góc tại tâm của robot.

(v_l, v_r) : Vận tốc tuyến tính của các bánh xe bên trái và bên phải.

v : Vận tốc tuyến tính tại tâm của robot.

ω : Vận tốc góc tại tâm của robot.

Dải đàn hồi (đại diện cho quỹ đạo chuyển động của robot) theo vật lý là đồng nhất, không thay đổi các đặc tính của nó trong quá trình tính toán quỹ đạo chuyển động của robot. Khi đó, đối với Elastic strips, dải đàn hồi này được gắn kết và tạo bởi một lưới liên kết hai chiều giữa các lò xo (Hình 2.9), ở đây minh họa cho một khớp cánh tay của robot và $E = (q_1, q_2, q_3)$ là các vị trí tương ứng tại thời điểm q_1, q_2, q_3 của robot trong quá trình di chuyển đến điểm đích.

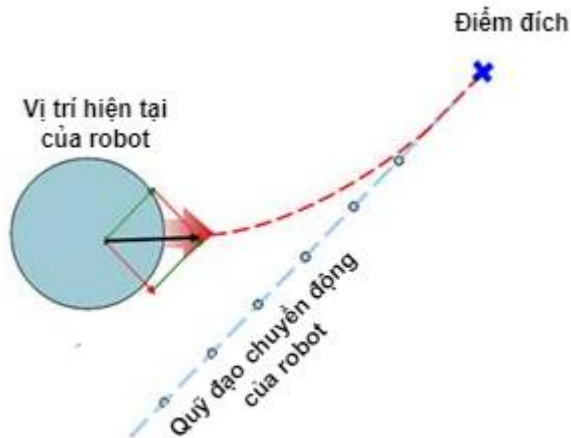


Hình 2.11. Cấu trúc của Elastic strips

2.3.1. Nội lực

Đây là một thành phần của Elastic strips được tạo ra bởi các liên kết đàn hồi tác động lên các robot, tương ứng tại thời điểm q_i sẽ xuất hiện điểm ${}^j P$ được gọi là điểm điều khiển của robot.

Tại bất kỳ vị trí nào trong quá trình di chuyển của robot, luôn có một véc tơ nội lực F^{int}_i kéo robot về điểm đích. Thành phần nội lực này là lực hấp dẫn tạo bởi điểm đích lên robot, véc tơ lực có hướng đến điểm đích và có xu hướng kéo robot về điểm đích tại mỗi thời điểm di chuyển (i) (Hình 2.12). Lực tác dụng tại điểm i lên khớp nối thứ j và điểm điều khiển ${}^j P$ của robot được xác định bởi công thức (2.16).



Hình 2.12. Nội lực

Để tính toán thành phần lực này, tại thời điểm lấy mẫu là $(t-1), (t), (t+1)$ tương ứng các điểm điều khiển ${}^{j-1}P, {}^jP, {}^{j+1}P$ (điểm điều khiển jP - tương ứng với điểm điều khiển tại thời điểm hiện tại; điểm ${}^{j-1}P$ - tương ứng với thời điểm $j-1$, điểm ${}^{j+1}P$ - tương ứng với thời điểm kế tiếp mà robot dự kiến sẽ thay đổi). Điểm i là trạng thái của robot tại thời điểm i (Hình 2.11) tương ứng là $q_i^{j-1}, q_i^j, q_i^{j+1}$.

$$F_i^{\text{int}} = k_c \left(\frac{{}^{j-1}d}{{}^{j-1}d + {}^j d} ({}^{j+1}P - {}^{j-1}P) - ({}^jP - {}^{j-1}P) \right) \quad (2.16)$$

Trong đó:

- ${}^{j-1}d$: khoảng cách giữa 2 điểm điều khiển ${}^{j-1}P, {}^jP$

$${}^{j-1}d = \sqrt{\left({}^j x - {}^{j-1} x \right)^2 + \left({}^j y - {}^{j-1} y \right)^2 + \left({}^j z - {}^{j-1} z \right)^2} \quad (2.17)$$

- ${}^j d$: khoảng cách giữa 2 điểm điều khiển ${}^jP, {}^{j+1}P$

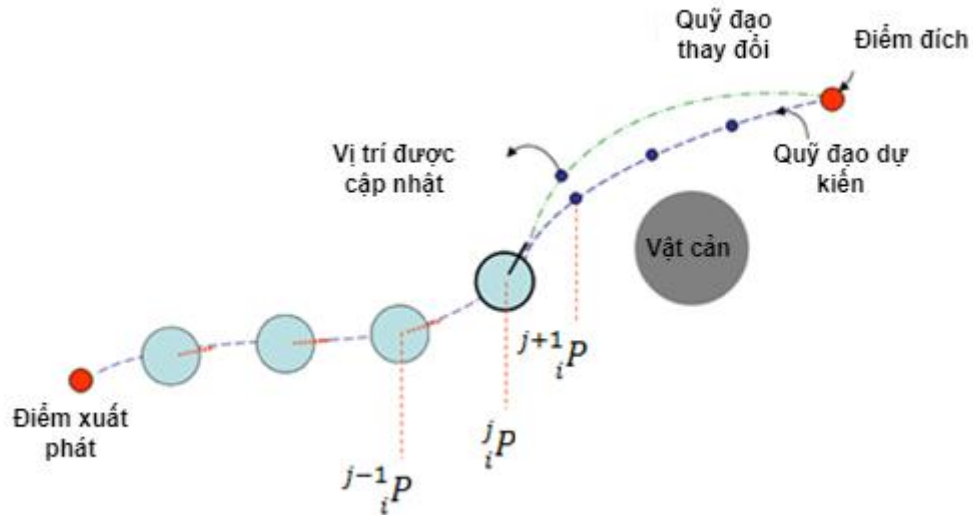
$${}^j d = \sqrt{\left({}^{j+1} x - {}^j x \right)^2 + \left({}^{j+1} y - {}^j y \right)^2 + \left({}^{j+1} z - {}^j z \right)^2} \quad (2.18)$$

- k_c : hệ số của nội lực.

- $\frac{d_i^{j-1}}{d_i^{j-1} + d_i^j}$: hệ số khoảng cách tương đối giữa các

trạng thái liên kế.

Các điểm ${}^{j-1}P_i, {}^jP_i, {}^{j+1}P$ được mô tả chi tiết trong hình 2.13, trong đó ${}^{j-1}P_i, {}^jP_i, {}^{j+1}P$ là điểm điều khiển của của robot tại thời điểm (i) với trạng thái (j).

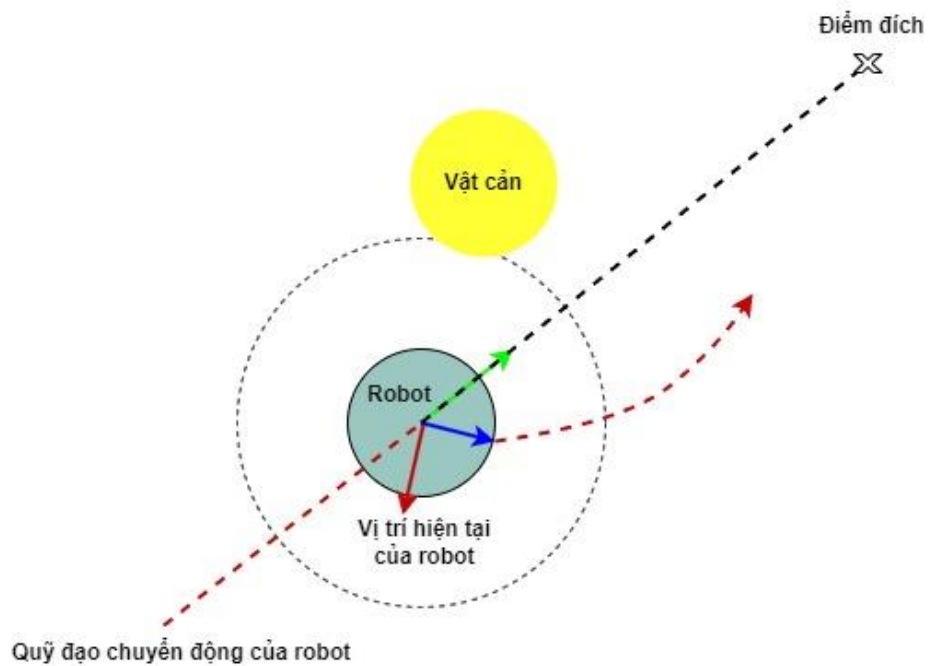


Hình 2.13. Sự thay đổi quỹ đạo chuyển động của robot [67]

Thành phần nội lực được quyết định bởi khoảng cách giữa các ${}^{j-1}P_i, {}^jP_i, {}^{j+1}P$ và véc tơ hướng đích. Nếu như khoảng cách giữa ${}^jP_i, {}^{j+1}P$ trở lên xa hơn, thì nội lực sẽ càng lớn và sẽ tác động lên quỹ đạo chuyển động của robot để tiến gần đến quỹ đạo trước đó. Nghĩa là không có sự dịch chuyển quá lớn về véc tơ di chuyển của robot tại thời điểm trước đó và thời điểm dự kiến di chuyển được xác lập sau khi tính toán.

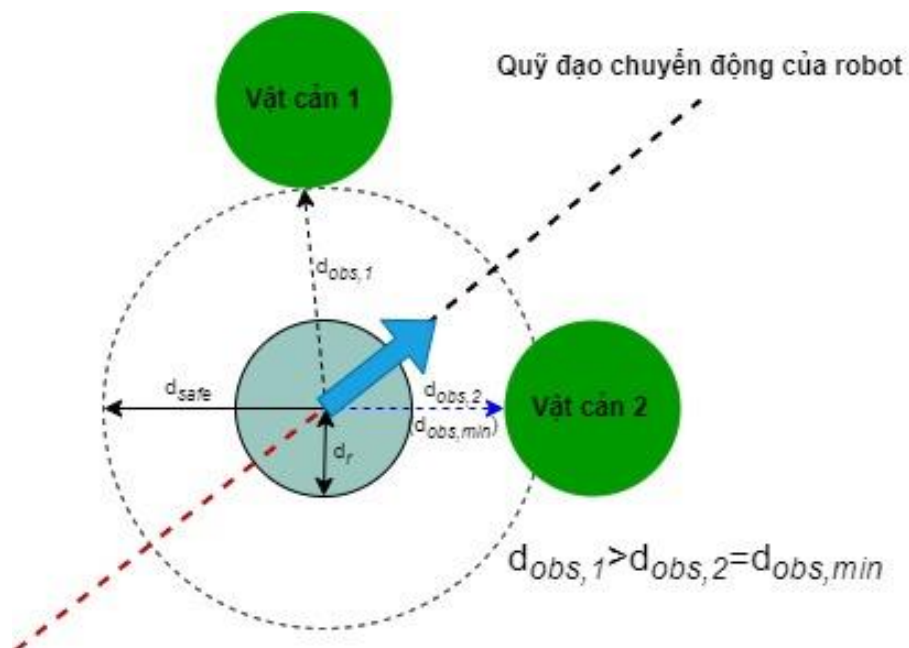
2.3.2. Ngoại lực

Thành phần thứ hai của dải đàn hồi được gọi là ngoại lực Ngoại lực (External force) là thành phần thứ hai của dải đàn hồi. Lực này được giả định là trường lực ảo được tạo bởi lực tác động từ vật cản lên robot với giả thiết có sự va chạm giữa robot với vật cản trong quá trình di chuyển. Giả thiết rằng khi di chuyển trong môi trường làm việc, khu vực nào đó robot sẽ va chạm với vật cản gần nó nhất, khi đó sẽ phát sinh một lực đẩy từ vật cản về phía robot nhằm mục tiêu đẩy robot ra xa vật cản đó và không xảy ra va chạm (Hình 2.14).



Hình 2.14. Ngoại lực

Độ lớn của thành phần ngoại lực được tính toán khi phát hiện xảy ra va chạm và được dựa khoảng cách ngắn nhất đến vật cản đến robot tại thời điểm xác định. Các dữ liệu khoảng cách được cung cấp bao gồm cả góc xác lập tương ứng giữa robot và vật cản, để từ đó việc tính toán véc tơ ngoại lực tác động lên robot sẽ được thực hiện.



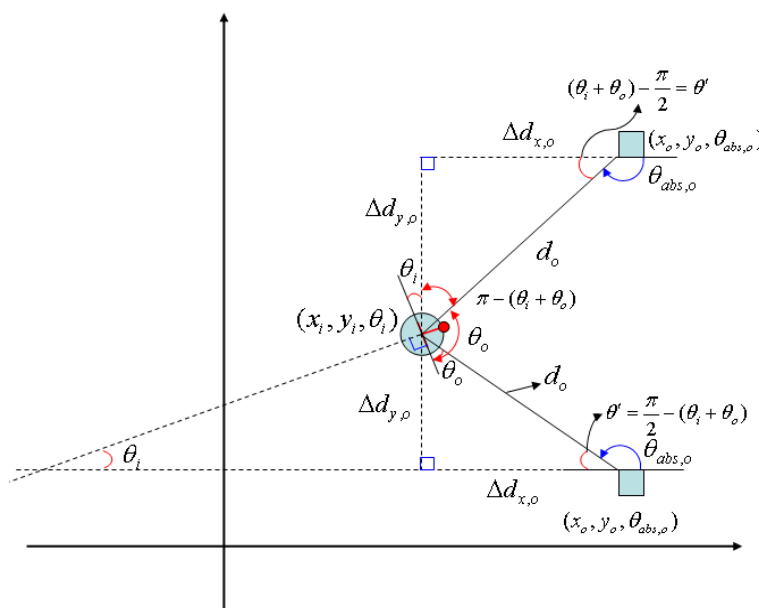
Hình 2.15. Không gian an toàn cho robot

Kĩ thuật này sử dụng bất kỳ khoảng cách để xác định việc có xảy ra va chạm giữa robot và vật cản hay không, gọi là khoảng cách an toàn (d_{safe}) (Hình 2.15). Để xác định khả năng phát sinh ngoại lực, ta tiến hành so sánh dữ liệu khoảng cách ngắn nhất từ vật cản đến robot với giá trị của d_{safe} . Nếu khoảng cách từ robot đến vật cản nhỏ hơn D_{safe} thì sẽ xảy ra va chạm và phát sinh ngoại lực.

Phương pháp này giả thiết không gian an toàn của robot là một hình tròn (2D) với bán kính là d_{safe} . Khi đó, nếu vật cản xâm phạm không gian an toàn, khoảng cách giữa điểm kiểm soát và vật cản gần nhất được xác định thông qua dữ liệu cảm biến hoặc một kĩ thuật nào đó, là cơ sở để tính toán độ lớn của thành phần ngoại lực.

Trong đó:

- d_{obs} : khoảng cách giữa robot và vật cản.
- d_r : bán kính của robot;
- d_{safe} : bán kính khoảng cách an toàn;
- d_o : khoảng cách sử dụng để tính toán ngoại lực;
- $d_{\text{obs, min}}$: khoảng cách ngắn nhất giữa robot và vật cản;
- d_o : được tính là $d_o = d_{\text{safe}} + d_r$;
- $d_{\text{obs, min}}$: là khoảng cách ngắn nhất giữa robot và các vật cản;
- $d_{\text{obs, i}}$: khoảng cách ngắn nhất giữa robot và vật cản i .



Hình 2.16. Tính toán vị trí vật cản

Việc tính toán vị trí vật cản trong tọa độ thực, ta sử dụng luật hình học cosin (Hình 2.16) với các tham số gồm vị trí và hướng của robot hiện tại; khoảng cách từ robot đến vật cản; góc giữa hướng robot và vật cản.

Giả sử $P_o=[x_o, y_o, z_o]^T$ là tọa độ của vật cản, $P_i=[x_i, y_i, z_i]^T$ là tọa độ của vị trí hiện tại của robot tại thời điểm (i). Vị trí (x_o, y_o) của vật cản với mỗi trục tại công thức (2.19).

$$\begin{cases} x_o = x_i + d_o \sin(\theta_i + \theta_o) \\ y_o = y_i + d_o \cos(\theta_i + \theta_o) \\ \theta_{obs,o} = -(\pi - \theta_i) \end{cases} \quad (2.19)$$

Sau khi xác định được khoảng cách ngắn nhất từ robot đến vật cản, ta tính được giá trị của thành phần ngoại lực theo phương pháp trường năng lượng được sẽ được áp dụng đối với điểm P theo công thức (2.20).

$$V_{ext}(P) = \begin{cases} \frac{1}{2} K_r (d_o - d_{obs,min})^2 & ; d_o > d_{obs,min} \\ 0 & ; d_o \leq d_{obs,min} \end{cases} \quad (2.20)$$

Tính toán gradient cho véc tơ lực $V_{ext}(P)$ tại điểm P:

$$F_i^{ext} = -\nabla V_{ext}(P) = K_r (d_o - d_{obs,min}) \frac{\vec{d}}{\|d\|} \quad (2.21)$$

$$\vec{d} = P_m - P_o; \|d\| = \sqrt{(x_m - x_o)^2 + (y_m - y_o)^2 + (z_m - z_o)^2} \quad (2.22)$$

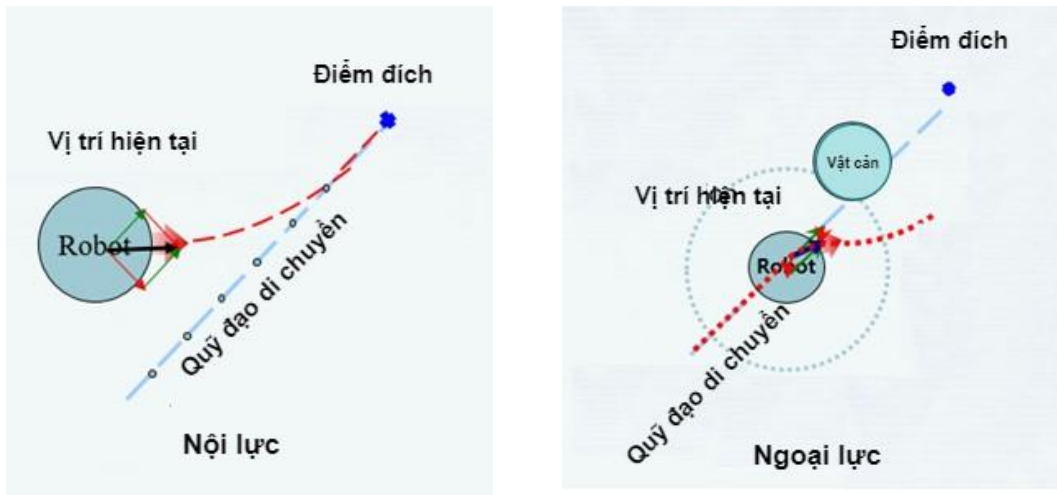
Trong đó:

- F_i^{ext} : lực đẩy từ phía vật cản lên robot;
- $d_{obs, min}$: khoảng cách ngắn nhất giữa robot đến vật cản;
- K_c : hệ số tính toán ngoại lực;
 K_c càng lớn, lực đẩy càng lớn;
- \vec{d} : véc tơ chỉ phương được xác lập tại điểm hiện tại P với điểm va chạm với vật cản gần nhất;
- Véc tơ \vec{d} được xác định và tính theo công thức (2.22);

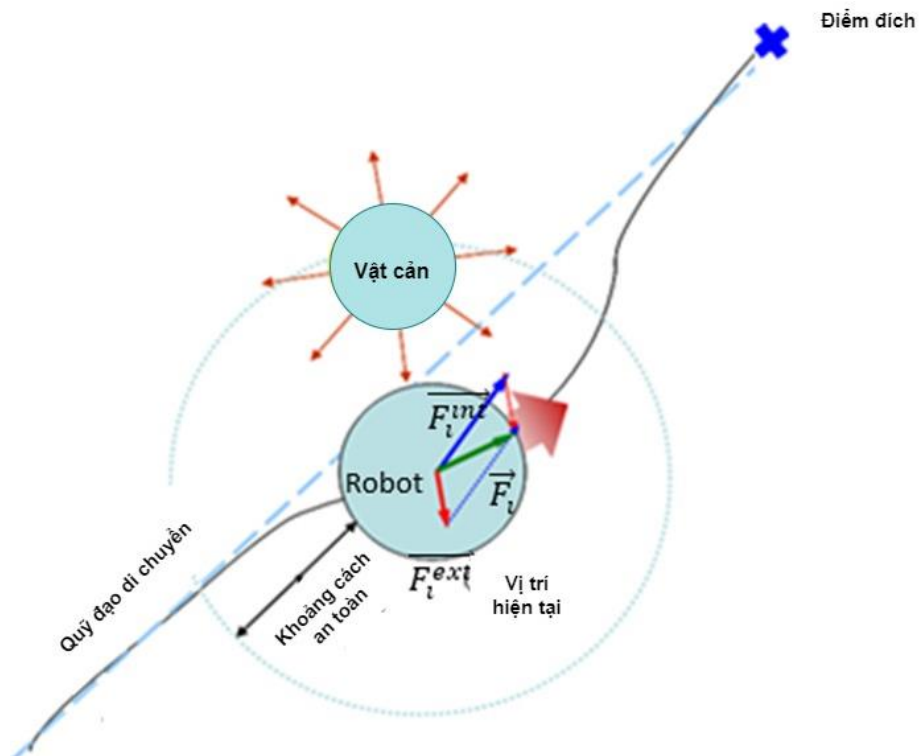
Với phương pháp này, không có ngoại lực đối với điểm bắt đầu và điểm kết thúc của quỹ đạo chuyển động của robot.

2.3.3. Tổng hợp lực theo phương pháp Elastic strips

Véc tơ tổng hợp lực đàn hồi Elastic strips F_i được tổng hợp từ hai thành phần lực là nội lực F_i^{int} và ngoại lực F_i^{ext} (Hình 2.18) và được tính theo (2.23).



Hình 2.17. Nội lực và Ngoại lực của Elastic strips



Hình 2.18. Tổng hợp lực Elastic strips

$$F_i = F_i^{\text{int}} + F_i^{\text{ext}}; F_i = [f_{i,x}, f_{i,y}, f_{i,z}]^T \quad (2.23)$$

2.3.4. Tính toán và cập nhật tọa độ cho robot tự hành

Để cập nhật vị trí di chuyển tại thời điểm (i) của robot, ta tiến hành cập nhật vị trí di chuyển của robot bằng cách tính toán, cập nhật tọa độ của robot theo các trục (x, y, z) và góc dịch chuyển (góc quay θ_i) của robot. Kết quả tính toán các tham số trên được mô tả tại (2.25) và (2.26). Trong đó:

$$\begin{cases} \Delta x = (x_i^{j+1} - x_{i,cur}^j) + f_{i,x} \\ \Delta y = (y_i^{j+1} - y_{i,cur}^j) + f_{i,y} \\ \Delta z = (z_i^{j+1} - z_{i,cur}^j) + f_{i,z} \end{cases} \quad (2.24)$$

Với: $(x_{i,cur}^j, y_{i,cur}^j, z_{i,cur}^j)$: vị trí hiện tại của robot;

$(x_i^{j+1}, y_i^{j+1}, z_i^{j+1})$: vị trí được xác lập theo quỹ đạo giả định ban đầu;

$(f_{i,x}, f_{i,y}, f_{i,z})$: giá trị của lực đàn hồi được tính toán;

Các tham số vị trí được cập nhật:

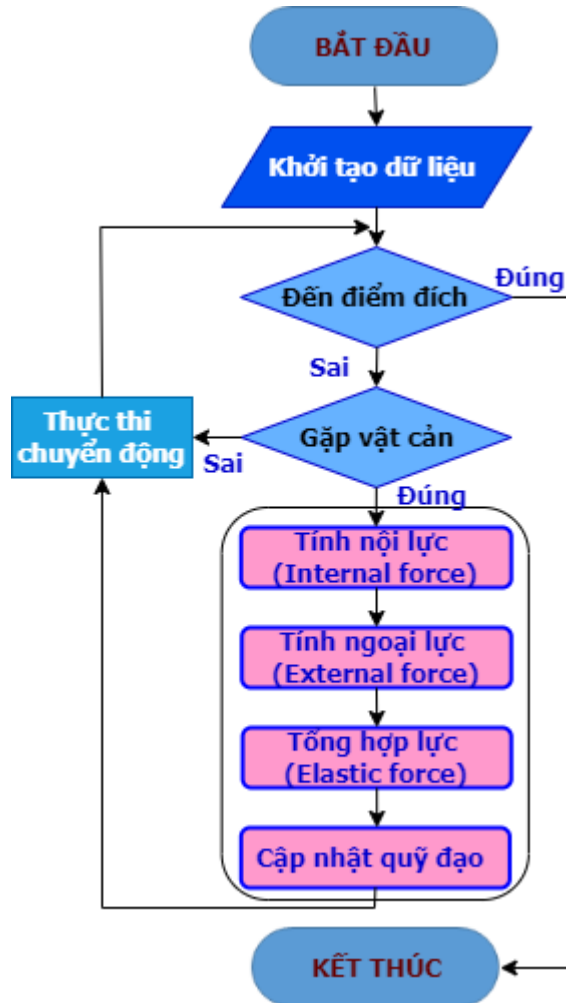
$$\begin{cases} x_{i,update} = \Delta x + x_{i,cur}^j \\ y_{i,update} = \Delta y + y_{i,cur}^j \\ z_{i,update} = \Delta z + z_{i,cur}^j \end{cases} \quad (2.25)$$

Tham số góc quay của robot:

$$\begin{cases} \Delta \theta_i = \arctan 2 \left(\frac{\Delta y}{\Delta x} \right) \\ \theta_{i,update} = \Delta \theta_i + \theta_{i,cur} \end{cases} \quad (2.26)$$

Với $\theta_{i,cur}$ là góc quay tại thời điểm hiện tại của robot; $\theta_{i,update}$ là góc quay mới được xác lập thông qua lực đàn hồi được tính toán.

2.3.5. Thuật toán tránh và chạm bằng phương pháp Elastic strips



Hình 2.19. Sơ đồ thuật toán tránh và chạm dựa trên Elastic strips

Thuật toán tránh và chạm dựa trên phương pháp Elastic strips [71] được mô tả tại hình 2.19.

Tại mỗi thời điểm (t) theo thời gian, các bước được thực hiện như sau:

1- Khởi tạo ban đầu là việc khởi gán các giá trị của thuật toán, thiết lập quỹ đạo di chuyển ban đầu cho robot (khởi tạo điểm bắt đầu, điểm kết thúc của quỹ đạo chuyển động, thông số của phương pháp Elastic strips (K_c , K_r), tham số di chuyển của robot (x_0 , y_0 , z_0 và θ_0).

2. Tiếp nhận dữ liệu để phân tích, tính toán và xác định việc xảy ra va chạm giữa robot và các vật cản, đồng thời cung cấp dữ liệu tính toán cho thuật toán Elastic strips để tính toán các thành phần lực cấu thành.

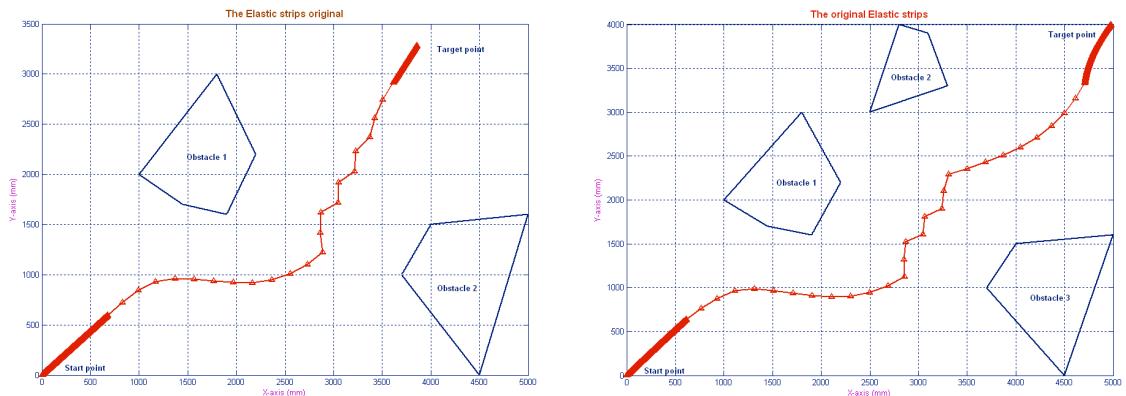
3. Elastic strips sẽ tính toán nội lực, ngoại lực và tổng hợp lực đàn hồi. Đồng thời, tính toán cập nhật tọa độ mới cho robot cũng như góc dịch chuyển để điều chỉnh quỹ đạo giúp robot tránh va chạm với vật cản.

4. Việc cuối cùng của robot là tiếp nhận dữ liệu vị trí, góc quay mới của robot, điều khiển hệ thống chấp hành để thực thi chuyển động cho robot.

2.2.2. Vấn đề tồn tại, hạn chế

Cùng với những ưu điểm vượt trội của phương pháp Elastic góc, tác giả Rostami [74] cũng nghiên cứu và chỉ ra một số hạn chế của phương pháp này khi triển khai, cài đặt cho robot tự hành:

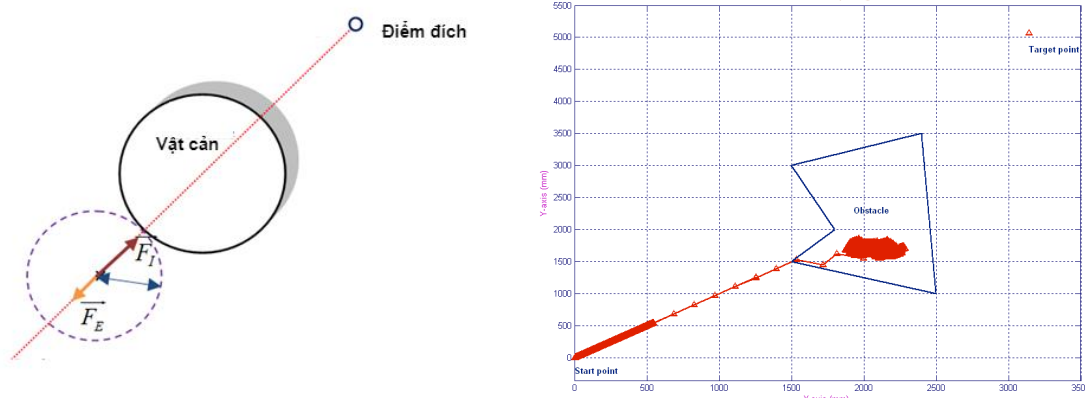
Một là, phương pháp góc đang tính toán ngoại lực dựa trên điểm duy nhất, gần nhất với chướng ngại vật. Ngoài ra, tại mỗi thời điểm thì việc tính toán lực đàn hồi Elastic strips, tính toán các vị trí di chuyển đề xuất cho robot, cập nhật vị trí cho robot là hoàn toàn độc lập nhau. Chính điều này dẫn đến việc, quỹ đạo chuyển động của robot sẽ không thực sự trơn hoặc thậm chí robot sẽ bị tình trạng quay trái và phải liên tục tạo thành quỹ đạo rích rích, gập khúc (Hình 2.20).



Hình 2.20. Quỹ đạo chuyển động của robot với phương pháp Elastic strips

Hai là, hướng di chuyển hiện tại của robot trùng với hướng của điểm va chạm gần nhất của vật cản. Khi đó, theo phương pháp này thì robot sẽ bị tắc vì phương của véc tơ nội lực và ngoại lực trùng nhau. Sẽ xảy ra 2 tình huống: 1- véc tơ nội lực lớn

hơn véc tơ ngoại lực, khi đó theo phương pháp gốc, robot phải đi lùi, điều này sẽ không áp dụng được trong mô hình thực tế và thời gian thực; 2- Nếu véc tơ nội lực nhỏ hơn véc tơ ngoại lực, khi đó tổng hợp véc tơ lực sẽ định hướng robot di chuyển thẳng, và lúc này sẽ xảy ra va chạm với vật cản (Hình 2.21).



Hình 2.21. Mô tả tổng hợp lực Elastic strips gốc khi hướng di chuyển hiện tại trùng với hướng điểm và chạm

Ba là, đối với phương pháp Elastic strips nếu như có tồn tại hơn 2 vật cản nằm về 2 phía trên đường di chuyển hiện tại của robot, khi tính toán theo phương pháp gốc, robot sẽ có trạng thái dịch chuyển về phía ngược lại do ngoại lực tác động dẫn đến việc tại bước dịch chuyển tiếp theo robot sẽ va chạm với vật cản ở phía còn lại.

Việc giải quyết những vấn đề tồn tại của phương pháp Elastic strips gốc là một yêu cầu thực tế để hoàn thiện hơn nữa thuật toán tránh vật cản, là cơ sở để nghiên cứu cải tiến phương pháp và thuật toán gốc cho robot tự hành để robot có thể tìm đường, tránh vật cản trên quỹ đạo chuyển động, hướng đến điểm đích sao cho quỹ đạo trơn mượt hơn, hiệu quả theo hệ thời gian thực, đáp ứng các yêu cầu thực tế. Đây là một trong những vấn đề nghiên cứu của luận án nhằm đề xuất một thuật toán cải tiến giúp cho robot tránh được vật cản trong quá trình làm việc.

2.3. Kỹ thuật phát hiện tránh va chạm dựa trên cấu trúc cây phân lớp hệ bao BVH, kết hợp với thuật toán Elastic strips

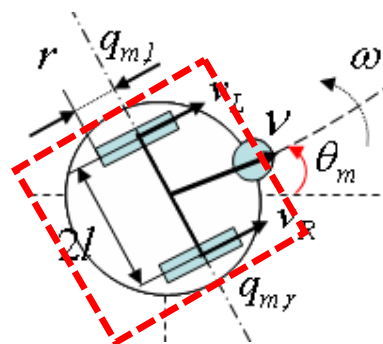
2.3.1. Đề xuất của phương pháp

Kỹ thuật tránh va chạm dựa trên cấu trúc cây phân lớp hệ bao BVH, kết hợp với thuật toán Elastic strips có cải tiến được dựa trên nguyên lý sử dụng kết hợp việc

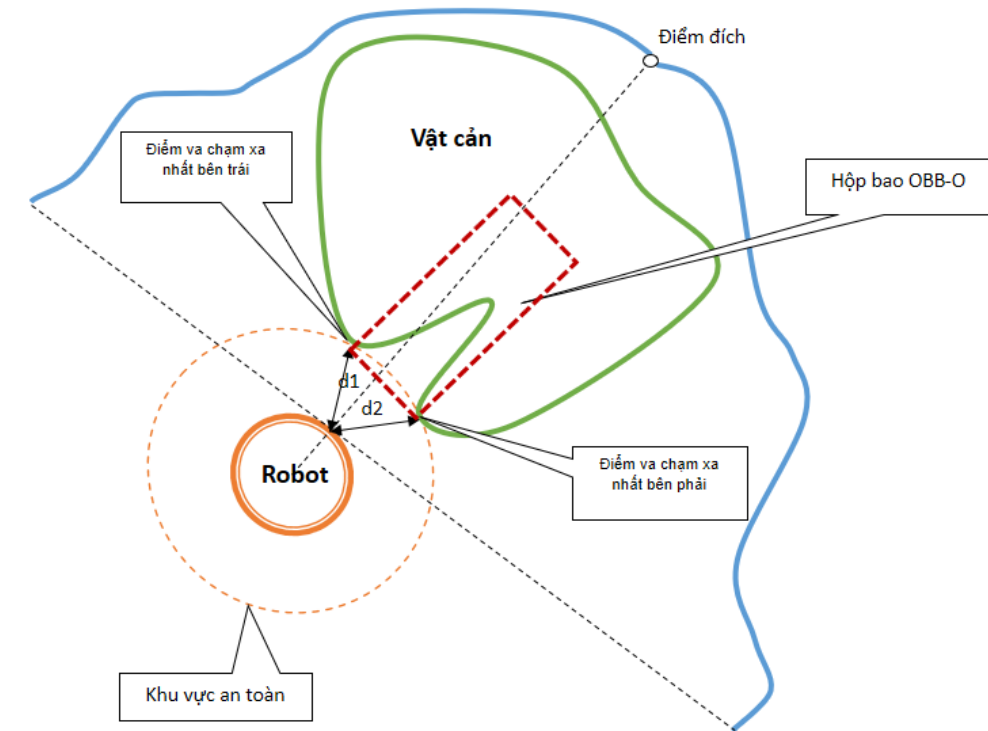
phát hiện, tính toán nhanh điểm va chạm sử dụng cây phân lớp hệ bao, kết quả tìm kiếm sẽ cung cấp chính xác điểm va chạm là dữ liệu đầu vào cho việc áp dụng thuật toán Elastic strips thực hiện việc cập nhật quỹ đạo cho robot để tránh vật cản trong quá trình di chuyển. Kỹ thuật kết hợp này ứng dụng hệ bao theo hướng và dựa trên các kết quả tính toán điểm va chạm với bảng dữ liệu có sẵn đảm bảo tính hiệu quả trong tính toán, đáp ứng thời gian thực [86].

Giả thiết robot và vật cản được bao bởi hai hộp bao theo hướng OBB là OBB-R và OBB-O, với OBB-R là hộp bao thiếu hướng của robot và OBB-O là hộp bao theo hướng của vật cản. Việc phát hiện va chạm giữa robot với vật cản lúc này được thay thế bằng việc xác định và tính toán điểm va chạm của hai hộp bao trên. Từ đó, tương ứng ta xây dựng được các cấu trúc cây phân lớp hệ bao BVH và tiến hành áp dụng phương pháp phát hiện, tính toán va chạm cho hai hệ bao đó.

Việc xác định hộp bao của robot (Hình 2.22) dựa trên kích thước thực tế của robot và phương dịch chuyển hiện tại, và hộp bao của vật cản dựa trên hai điểm gần nhất, một bên trái, một bên phải và dịch chuyển tịnh tiến khoảng cách tương ứng bằng với độ dài của d_{safe} theo phương robot đang di chuyển để xác định thêm hai điểm mốc của hộp bao vật cản (Hình 2.23).



Hình 2.22. Hộp bao của robot OBB-R



Hình 2.23. Hộp bao của vật cản OBB-O

Tại mỗi thời điểm di chuyển, với các thông tin nhận được từ hệ thống cảm biến sẽ xác định việc có xuất hiện chướng ngại vật hay không. Nếu chỉ có một điểm va chạm xảy ra, thuật toán không cần xây dựng hộp bao các đối tượng mà áp dụng thuật toán Elastic strip gốc để tính toán và xác định vị trí, tọa độ di chuyển tiếp theo cho robot một cách thông thường. Ngược lại, ta sẽ tiến hành lựa chọn hai điểm va chạm theo hướng di chuyển hiện tại (một bên trái, một bên phải) (hình 2.23) và xây dựng hộp bao của vật cản (OBB-O).

Thuật toán xây dựng hộp bao của vật cản:

Khởi tạo dữ liệu	<p>Mảng A: dữ liệu đọc vào từ cảm biến là khoảng cách đến các vật cản</p> <p>d_safe: Bán kính vùng an toàn của robot</p> <p>min_L: min_R: Khoảng cách ngắn nhất đến vật cản bên trái, phải</p>
Xác định khoảng cách ngắn nhất đến	<p>For $i=1$ to size of (A) do</p> <p>if $A [i] < d_safe$ then</p> <p>If $i \leq size(A) \div 2$ and $A [i] < min_L$ then</p>

vật cản từ 02 phía trái, phải của robot	<pre> min_L=A [i] end if If i> size of(A) div 2 and A [i]<min_R then min_R=A [i] end if end if end for if min_L=d_safe and min_R=d_safe then RETURN FALSE </pre>
Tạo hộp bao cho vật cản	<p><i>Xác định hộp bao theo hướng cho vật cản dựa trên giá trị tìm được của min_L, min_R</i></p> <p><i>Xác định hộp bao theo hướng cho robot dựa trên định nghĩa hoặc kích thước của robot</i></p>

Sau khi xây dựng được hộp bao cho vật cản và cho robot, việc xây dựng hệ bao cho B1, B2 và áp dụng phương pháp phát hiện, tính điểm va chạm sử dụng hệ bao giúp xác định được khả năng xảy ra va chạm và chính xác điểm va chạm giữa hai hệ bao, cũng chính là của robot và các vật cản.

Cùng với đó, ý tưởng cải tiến thuật toán Elastic strips gốc để giải quyết một số hạn chế đã được chỉ ra, giúp robot di chuyển hiệu quả hơn được đề xuất đó là: thay vì chỉ dựa vào duy nhất một điểm va chạm gần nhất với vật cản để tính toán thành phần ngoại lực, phương pháp cải tiến sẽ tính toán thành phần ngoại lực dựa vào hai thành phần: một bên trái (*dựa vào điểm gần nhất bên trái*), một bên phải (*dựa vào điểm gần nhất bên phải*) theo hướng di chuyển hiện tại của robot (Hình 2.24).

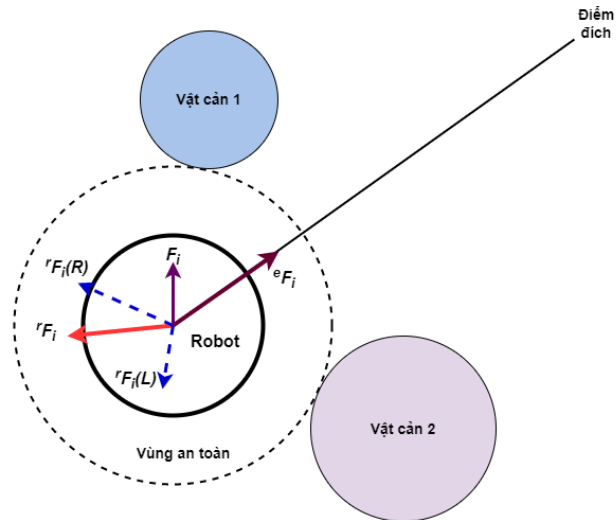
Thành phần ngoại lực gồm: ngoại lực trái (L) và phải (R) theo công thức (2.27):

$$\begin{aligned}
 V_{ext_L}(PL) &= \begin{cases} \frac{1}{2} K_r (d_{safe} - d_{obs,minL})^2 & ; d_{safe} > d_{obs,minL} \\ 0 & ; d_{safe} \leq d_{obs,minL} \end{cases} \\
 V_{ext_R}(PR) &= \begin{cases} \frac{1}{2} K_r (d_{safe} - d_{obs,minR})^2 & ; d_{safe} > d_{obs,minR} \\ 0 & ; d_{safe} \leq d_{obs,minR} \end{cases}
 \end{aligned} \tag{2.27}$$

Lực tổng hợp được xác định tại công thức (2.28):

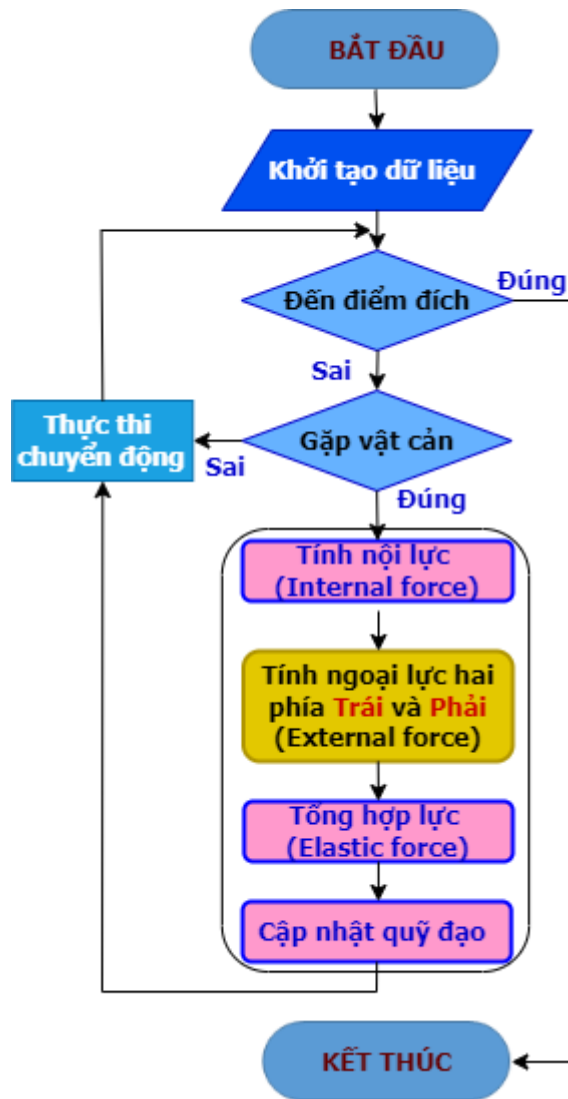
$$F_i = F_i^{int} + F_i^{ext}(L) + F_i^{ext}(R);$$

$$F_i = [f_{i,x}, f_{i,y}, f_{i,z}]^T \quad (2.28)$$



Hình 2.24. Xác định thành phần ngoại lực của phương pháp cải tiến

Đối với thành phần nội lực, việc xác định và tính toán vẫn theo thuật toán gốc và áp dụng các công thức (2.16),(2.21),(2.23).



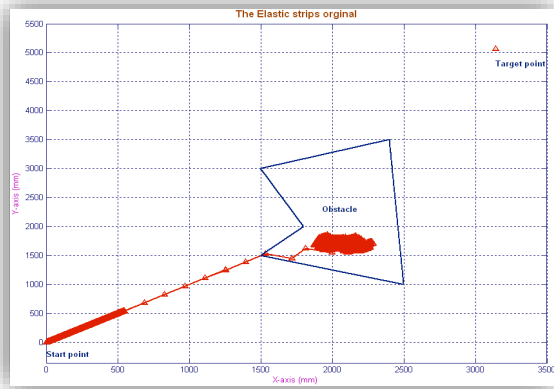
Hình 2.25. Sơ đồ thuật toán cải tiến Elastic strips

2.3.2. Kết quả tính toán và mô phỏng

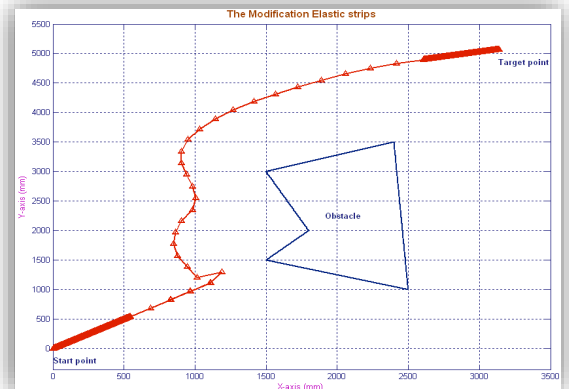
Bảng 2.3. Bảng dữ liệu mô phỏng cho robot

	Trường hợp 1	Trường hợp 2	Trường hợp 3	Trường hợp 4
Vị trí ban đầu của robot	(0, 0);	(0,0);	(0,0)	(0,0)
Tọa độ đích đến của robot	(3500,5000)	(4000,3500)	(5000,4000)	(6000,5500)
Hệ số nội lực K_c	0.8	0.8	0.6	0.6
Hệ số ngoại lực K_r	0.6	0.6	0.3	0.3
Bán kính vùng an toàn của robot	1000 (mm)	1000 (mm)	1000 (mm)	1000 (mm)
Bán kính robot	350 (mm)	350 (mm)	350 (mm)	350 (mm)

Với số liệu đầu vào theo bảng 2.3, kết quả mô phỏng đã được kiểm chứng và chứng minh tính đúng đắn của phương pháp cải tiến.



(a) Kết quả của thuật toán gốc

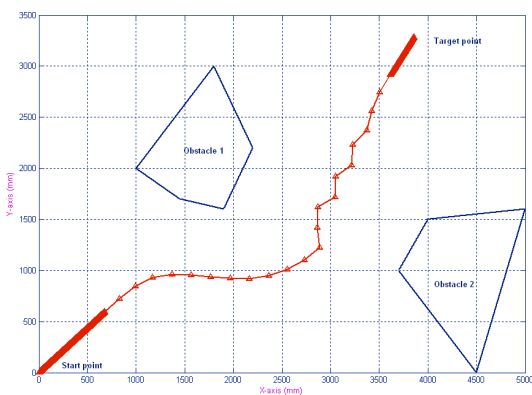


(b) Kết quả của thuật toán cải tiến

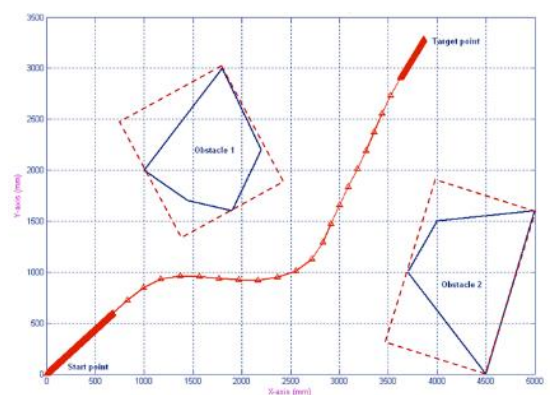
Hình 2.26. Kết quả mô phỏng trong trường hợp 1

Trong trường hợp này, ở phương pháp gốc robot không thể di chuyển đến được điểm đích và va chạm với vật cản khi hướng véc tơ ngoại lực trùng với véc tơ nội lực hướng đích của robot (Hình 2.26a). Tuy nhiên, khi cài đặt với thuật toán cải tiến, mặc dù quỹ đạo di chuyển không được mịn, nhưng robot tránh được va chạm và hướng đến đích (Hình 2.26b).

Trong các trường hợp (2),(3),(4) kết quả mô phỏng tại hình 2.27 và hình 2.28 cho thấy, cả hai phương pháp (gốc, cải tiến) đều đưa ra cho robot một quỹ đạo chuyển động và tránh được vật cản, hướng đến đích. Tuy nhiên, với phương pháp cải tiến quỹ đạo của robot mịn hơn, trơn hơn so với quỹ đạo của phương pháp gốc.



(a) Kết quả của thuật toán gốc



(b) Kết quả của thuật toán cải tiến

Hình 2.27. Kết quả mô phỏng trong trường hợp 2,3,4 với hai vật cản

CHƯƠNG 3. KỸ THUẬT TRÁNH VA CHẠM DỰA TRÊN TÍNH TOÁN VÀ PHÂN VÙNG ĐỒNG MỨC XÁC SUẤT VA CHẠM

Chương 3 đề xuất một kỹ thuật và thuật toán tránh va chạm cho robot dựa trên tính toán xác suất va chạm, phân vùng đồng mức xác suất va chạm. Các mô hình tính toán xác suất va chạm, phân bù xác suất va chạm được đề xuất.

3.1. Xác suất va chạm và ứng dụng

3.1.1. Khái niệm

Lý thuyết xác suất va chạm là một nhánh của toán học và vật lý xử lý việc tính toán và phân tích xác suất xảy ra va chạm giữa các đối tượng hoặc hạt, được ứng dụng trong nhiều lĩnh vực khác nhau bao gồm vật lý, kỹ thuật, khoa học máy tính và thống kê. Trong vật lý hạt nhân, lý thuyết xác suất va chạm được sử dụng để nghiên cứu và dự đoán kết quả của các va chạm hạt có năng lượng cao, chẳng hạn như trong các máy gia tốc hạt như máy gia tốc hạt lớn (LHC). Trong lĩnh vực cơ học quỹ đạo, lý thuyết xác suất va chạm được sử dụng để đánh giá nguy cơ va chạm giữa vệ tinh, tàu vũ trụ và mảnh vỡ không gian trong quỹ đạo Trái đất, bằng cách phân tích quỹ đạo và đặc điểm của các vật thể. Ngoài ra, lý thuyết xác suất va chạm được áp dụng trong mạng máy tính, đặc biệt là trong các giao thức như Ethernet nhằm tối ưu hóa giao thức và xác định xác suất va chạm gói tin, giúp giao tiếp hiệu quả và giảm thiểu mất dữ liệu hay trong kỹ thuật giao thông để đánh giá nguy cơ va chạm giữa các phương tiện tại các giao lộ hoặc trên đường giao thông thông qua việc xem xét các yếu tố như tốc độ xe, lưu lượng giao thông và hình học đường... Những năm gần đây, lý thuyết xác suất cũng được sử dụng trong các ngành công nghiệp để đánh giá rủi ro và đánh giá an toàn của các hệ thống, hay giữa các chất nguy hiểm.

Khái niệm xác suất va chạm được sử dụng để đánh giá và quản lý nguy cơ va chạm giữa các đối tượng hoặc thực thể trong cùng một môi trường hoạt động có thể là dưới nước, trên mặt đất, ngoài vệ tinh hay thậm chí trong môi trường vật lý hạt

nhân. Bao gồm: va chạm giữa các hạt trong vật lí, va chạm giữa các tàu biển, phương tiện giao thông, các tàu vũ trụ, vệ tinh...

Những năm gần đây, cùng với sự phát triển của một số ngành dịch vụ như bảo hiểm, logistic, xe tự lái... thì xác suất va chạm được sử dụng để đánh giá rủi ro và bảo lãnh phát hành. Các công ti bảo hiểm phân tích dữ liệu lịch sử, mô hình lái xe và các yếu tố liên quan khác để ước tính khả năng va chạm hoặc tai nạn cho các chủ hợp đồng cá nhân hoặc các khu vực cụ thể. Vì thế, phân tích xác suất va chạm là yếu tố quan trọng của công nghệ xe tự lái, nhằm hỗ trợ việc ra quyết định di chuyển trong thực tế.

Việc tính toán, xác định xác suất va chạm cũng như phân tích xác suất va chạm để đánh giá rủi ro, đưa ra quyết định đúng đắn và thực hiện biện pháp phòng ngừa thích hợp để giảm nguy cơ va chạm đang là một vấn đề được quan tâm và nghiên cứu những năm gần đây.

3.1.2. Ứng dụng

3.1.2.1. Phân tích độ nhạy của xác suất va chạm

Phân tích độ nhạy là một kỹ thuật được sử dụng để hiểu tác động của những thay đổi trong các biến đầu vào đối với đầu ra của mô hình hoặc phân tích. Đối với xác suất va chạm, phân tích độ nhạy giúp xác định được những yếu tố có ảnh hưởng đến xác suất va chạm và độ chính xác của kết quả tính toán [56].

Phân tích độ nhạy là một thành phần của phân tích xác suất, để xác định độ nhạy của xác suất tính toán đối với các biến thể dự kiến trong các tham số mô tả tính toán [18]. Ảnh hưởng của các yếu tố như hình học kết hợp, hiệp phương sai vị trí, kích thước của đối tượng đến tính toán P_c sẽ được xem xét để phân tích, đánh giá độ nhạy của P_c .

Phân tích độ nhạy được chia thành hai phương pháp cơ bản: 1- Phương pháp phân tích nhằm tính toán đạo hàm riêng của các biến độc lập dựa trên biểu thức phân tích của các biến phụ thuộc để được biểu thức phân tích về độ nhạy xác suất va chạm. 2- Phương pháp số nhằm thay đổi giá trị biến độc lập để thay đổi giá trị biến

phụ thuộc và từ đó có được độ nhạy với một số quan hệ hàm phức tạp, các biểu thức phân tích.

3.1.2.2. Xác suất va chạm tối đa

Đối với khoảng cách trượt và kích thước của các vật thể không gian, có xác suất va chạm tối đa (P_{cmax}) thay đổi theo kích thước, hình dạng và hướng của ellipsoid hiệp phương sai lỗi. Ước tính P_{cmax} là quan trọng trong đánh giá rủi ro kết hợp khi một hoặc cả hai đối tượng không được biết hoặc không đáng tin cậy, P_{cmax} như một chỉ số về rủi ro để đánh giá sự kết hợp trong trường hợp xấu nhất hay được sử dụng trong bộ lọc trước của đối tượng nguy hiểm nếu thời gian tính toán cho P_{cmax} nhỏ hơn đáng kể so với tính toán của P_c .

Xác suất va chạm tối đa P_{cmax} được tính toán nhanh chóng bằng các biểu thức phân tích hoặc gần đúng, trong khi phân tích xác suất va chạm được xác định tốn nhiều thời gian hơn như tích phân hoặc lặp lại chuỗi và P_{cmax} thường lớn hơn rủi ro thực tế. P_{cmax} và phương sai sai số tương ứng có thể xác định độ chính xác dự đoán quỹ đạo cần thiết để đánh giá kết hợp sẽ ngăn ngừa hoặc giảm thiểu độ sai lệch của các tính toán xác suất [4, 32, 69].

Có nhiều nghiên cứu về xác suất va chạm tối đa được thực hiện dựa trên các ứng dụng khác nhau: Alfriend [5] giả định rằng hàm mật độ xác suất là không đổi trên hình cầu và bằng giá trị ở trung tâm của hình cầu. Dựa trên giả định này, giá trị của hệ số chia tỉ lệ hiệp phương sai tối đa hóa P_c và giá trị tương ứng của P_{cmax} được trình bày. Alfano [4] phát triển một phương pháp để lập bản đồ các vùng có xác suất tối đa cho kích thước các vệ tinh, tiếp cận kích thước và hình dạng hiệp phương sai. Các biểu đồ được tạo ra cho thấy tác động của sự không chắc chắn về vị trí đối với tính toán xác suất và đánh giá mật độ xác suất. Alfano [3] đã chỉ ra cách tính toán các giới hạn trên của xác suất bằng cách ngăn chặn việc khai thác các thông số và hướng hiệp phương sai "tồi tệ nhất" theo một số giả định cực đoan khi tỉ lệ khung hình của hiệp phương sai kết hợp tiếp cận độ chính xác. Chan [14] đã thảo luận hai cách tiếp cận để xác định xác suất tối đa: phương pháp xác suất hợp lệ cực đại và phương pháp xác suất tối đa. Cách tiếp cận đầu tiên được sử dụng để có

được ước tính nhanh về xác suất tối đa khi hàm mật độ xác suất không thay đổi nhiều, tương tự như cách xác định từ Alfriend [5]. Cách tiếp cận thứ hai là phân tích chính xác vấn đề phụ thuộc vào độ chính xác của biểu thức phân tích với xác suất va chạm, các biến được xác định và phân tích là kích thước và hình dạng. Laporte [53] đã trình bày một phương pháp xác suất tối đa bằng cách chia tỉ lệ từng hiệp phương sai riêng lẻ để tìm ra xác suất tối đa cho hình dạng và hướng cố định của các elip lỗi. Frisbee [28] đã nghiên cứu cách tiếp cận xác suất tối đa nếu thông tin về hiệp phương sai lỗi chỉ dành cho một trong hai đối tượng, thông qua thông tin hiệp phương sai đã biết cùng với giá trị tới hạn của hiệp phương sai bị thiếu để thu được cận trên P_c gần đúng.

Ngoài ra, cần lưu ý rằng P_{cmax} có khác biệt đáng kể so với P_c thực tế với một tập hợp dữ liệu nhất định và P_{cmax} khác nhau theo thứ tự độ lớn so với giá trị thực tế được tính trực tiếp từ dữ liệu. Nếu P_{cmax} nhỏ hơn mức chấp nhận được, không cần thực hiện phân tích xác suất và định nghĩa ngưỡng chấp nhận của P_{cmax} tùy thuộc vào người dùng, yêu cầu. Để tránh bỏ lỡ cảnh báo, ngưỡng P_{cmax} phải nhỏ hơn ngưỡng P_c nhỏ nhất (ví dụ ngưỡng màu vàng) vì nếu P_{cmax} lớn hơn ngưỡng chấp nhận thì sẽ không có ý nghĩa gì vì khi đó P_c hoàn toàn được tính toán bằng thực tế.

3.1.2.3. Phân tích Thiểu và Báo động Sai

Phân tích thiểu và báo động sai là một thành phần quan trọng của đánh giá kết hợp trong việc ứng dụng của xác suất va chạm. Đánh giá sự kết hợp đề cập đến quá trình dự đoán và phân tích các phương pháp tiếp cận gần hoặc va chạm tiềm năng giữa các vật thể không gian, chẳng hạn như vệ tinh hoặc mảnh vụn không gian.

Xác suất va chạm P_c giữa hai đối tượng được xác định dựa trên thông tin các vector vị trí, vận tốc và phương sai lỗi liên quan đến phương pháp TCA. Đây là một trong những kỹ thuật chính trong đánh giá rủi ro kết hợp. P_c liên quan chặt chẽ đến các vector vị trí và vận tốc, hiệp phương sai liên quan và bán kính kết hợp của hai không gian. Những thông tin này thu được thông qua các quan sát, ước tính hoặc tính toán. Sự không chắc chắn vốn có của các thông tin này sẽ ảnh hưởng đến mức độ tin cậy của đánh giá rủi ro kết hợp.

Trong đánh giá kết hợp, một cảnh báo bị thiếu xảy ra khi một kết hợp nguy hiểm tiềm tàng giữa hai đối tượng không được phát hiện hoặc xác định. Thiếu báo động có thể xảy ra do các yếu tố khác nhau, bao gồm cả những hạn chế trong vùng phủ sóng cảm biến, khoảng cách dữ liệu, lỗi trong dữ liệu theo dõi, hoặc không chính xác trong các mô hình xác suất va chạm. Và vì thế, phân tích thiếu cảnh báo liên quan đến việc kiểm tra dữ liệu lịch sử để xác định các trường hợp mà các phương pháp tiếp cận gần hoặc va chạm bị bỏ qua giúp cải thiện độ chính xác và độ tin cậy của các mô hình xác suất va chạm.

Báo động sai trong đánh giá kết hợp xảy ra khi một cách tiếp cận gần hoặc va chạm được dự đoán hoặc xác định nhưng thực chất là một báo động sai. Báo động sai có thể phát sinh do sự không chắc chắn trong việc xác định quỹ đạo ban đầu, lỗi trong tính toán xác suất va chạm hoặc không chính xác trong dữ liệu cảm biến. Việc phân tích báo động sai liên quan đến việc xem xét các dự đoán trong quá khứ và báo động sai để xác định nguyên nhân và nguồn gốc của báo động sai giúp tinh chỉnh các thuật toán đánh giá kết hợp, cập nhật các quy trình xác định quỹ đạo và nâng cao độ tin cậy của các dự đoán va chạm.

3.1.2.4. Đánh giá toàn diện rủi ro va chạm

Đánh giá rủi ro va chạm về cơ bản là một sự chuyển đổi từ các tham số kết hợp dự đoán sang thông tin rủi ro hỗ trợ việc ra quyết định. Đây là một quá trình phân tích định lượng về rủi ro va chạm của sự kiện và định lượng độ tin cậy của rủi ro.

Xác suất va chạm giữa hai vật thể được ứng dụng để đánh giá rủi ro va chạm, giá trị xác suất này được xác định bởi các vector vị trí và vận tốc tương đối, hiệp phương sai vị trí tương đối và bán kính kết hợp của hai vật thể. Các tham số trên được xác định thông qua đo lường, ước tính và tính toán dựa từ đó tính toán xác định được xác suất va chạm P_c , bởi vậy nên độ chính xác của các tham số này ảnh hưởng rất lớn đến độ tin cậy của P_c .

3.2. Kỹ thuật phát hiện, tránh va chạm cho robot dựa trên tính toán xác suất

Trong lý thuyết về tự động hóa, các nghiên cứu xác định đường di chuyển của robot tập trung giải quyết vấn đề giảm thời gian và quãng đường di chuyển cho robot

mà không được va chạm với các vật cản. Trong trường hợp tổng quát thì các vật cản cũng là các đối tượng cùng tham gia di chuyển theo một quỹ đạo được xác định sẵn hoặc ngẫu nhiên.

Tránh va chạm dựa trên phản ứng là một phương pháp để xử lý va chạm trong hệ thống tự động và robot. Phương pháp này dựa trên việc sử dụng các thuật toán để xác định khi một đối tượng gặp vật cản và thực hiện các hành động tránh vật cản đó. Các thuật toán tránh va chạm bao gồm các thuật toán tìm đường tránh, tìm đường đi ưu tiên và thuật toán tính toán vận tốc và gia tốc để tránh va chạm. Đây đang được coi là một phương pháp hiệu quả cho việc giải quyết tránh va chạm trong hệ thống tự động, robot trong môi trường động, thời gian thực và có sự tham gia của nhiều robot, nhiều vật cản.

Do đó, để tối ưu về đường đi, ý tưởng đề xuất là xác định đường đi xuyên qua vùng có khả năng xảy ra va chạm trong đó lựa chọn những đường đi qua vùng có xác suất xảy ra va chạm thấp để xây dựng quỹ đạo chuyển động cho robot. Khi đó, vấn đề đặt ra là xác định xác suất va chạm giữa các đối tượng trong không gian quan sát, đây là bài toán quan trọng mang tính chất quyết định cho việc xây dựng quỹ đạo tối ưu cho robot.

Nghiên cứu tính toán xác suất va chạm giữa các đối tượng được ứng dụng không chỉ giải quyết vấn đề tránh va chạm (các tàu vận tải, thiết bị tự lái, robots,...) mà còn được vận dụng trong khoa học quân sự: tính toán điểm, vùng va chạm (đánh chặn, tính toán điểm mục tiêu của vũ khí và tên lửa,..). Các nghiên cứu này đề cập đến việc sử dụng các mô hình xác suất để xác định hai đối tượng có khả năng xảy ra va chạm (xác suất >0), dờn nhau (xác suất $=0$). Các mô hình xác suất va chạm thường được xây dựng dựa trên dữ liệu về vận tốc, gia tốc và vị trí của đối tượng.

Đối với bài toán phát hiện tránh va chạm dựa trên xác suất cho robot, các nhà nghiên cứu tập trung vào việc phát hiện, tránh va chạm giúp robot di chuyển trong môi trường mà không gây va chạm với vật cản bao gồm: chướng ngại vật, người điều khiển và các robot khác cùng làm việc.

Đối với bài toán đánh chặn và phòng thủ, việc ứng dụng xác suất va chạm là một yếu tố quan trọng giúp đo lường xác suất tấn công xảy ra và đề xuất phương án đánh chặn hoặc giảm thiểu mức độ rủi ro, định lượng mức độ, nguy cơ của các tấn công; xác định và đánh giá các yếu tố gây nguy cơ cho hệ thống phòng thủ bao gồm cả nguồn tấn công và mức độ tấn công; thiết kế và đánh giá các giải pháp phòng thủ.

Hầu hết các nghiên cứu phát hiện, tránh va chạm dựa trên xác suất tập trung vào trạng thái hiện tại của đối tượng và tính toán nguy cơ va chạm dựa trên phương pháp Velocity Obstacle (VO), quỹ đạo chuyển động được xác định bằng việc tránh các vùng có khả năng va chạm, nên đường đi của robot là không tối ưu về quãng đường, tốn kém thời gian.

Trong những năm gần đây, một số phương pháp tránh va chạm đã được nghiên cứu dựa trên xác suất va chạm, chẳng hạn như: Kuwata và cộng sự [50] đã sử dụng phương pháp Velocity Obstacles (VO) để thực hiện các mô phỏng tránh va chạm cho thiết bị không người lái (UAV); Kim [43] và Chang [80] đã thực hiện các nghiên cứu sử dụng mô hình Markov Decision Process (MDP) để tránh va chạm dựa trên mô hình ra quyết định Markov; Woo và cộng sự [78] đã thử nghiệm thuật toán tránh va chạm dựa trên kỹ thuật học máy tăng cường theo mô hình Semi-Markov Decision Process (SMDP). Các phương pháp này đều sử dụng xác suất để xác định và tính toán rủi ro va chạm hoặc ước tính các trạng thái không rõ ràng của môi trường.

Chen và cộng sự [15] đã nghiên cứu một phương pháp bằng cách sử dụng logic mờ sử dụng lý thuyết thời gian và khoảng cách đến điểm tiếp cận gần nhất (TCPA, DCPA) để nhận biết tránh va chạm; Park và cộng sự [68] dựa vào fuzzy logic đã tính toán rủi ro va chạm sử dụng TCPA và DCPA là hàm thành viên, kết hợp với đánh giá chỉ số rủi ro va chạm dựa trên áp lực từ môi trường, định lượng nguy cơ rủi ro va chạm trong các tình huống và so sánh với chỉ số rủi ro va chạm dựa trên logic mờ; Namgung và cộng sự [64, 65] sử dụng neuro fuzzy để tính toán chỉ số rủi ro va chạm (CRI). Nghiên cứu này xem xét thêm các hệ số động từ vật cản; Fulgenzi và cộng sự [29, 30] thực hiện các mô phỏng tránh va chạm trong môi trường động (dynamic

enviroment) bằng cách kết hợp các lưới công suất động được cung cấp bởi cảm biến và xác suất vận tốc của vật cản (PVO).

Một thuật toán tránh va chạm dựa trên xác suất được đề xuất bởi Kim và cộng sự [44] đã sử dụng bộ lọc Kalman để ước tính các biến trạng thái của tàu và tính toán xác suất, dự đoán vị trí thông qua biến trạng thái và phân phối xác suất. Thuật toán đề xuất đã dự đoán các trạng thái và hoạch định đường đi tối ưu bằng cách xem xét các biến trạng thái của đối tượng, được tính toán thông qua bước dự đoán của bộ lọc UKF (Unscented Kalman filter).

3.2.1. Cơ sở toán học

3.2.1.1. Các định nghĩa

Dưới đây, là một số các định nghĩa đánh giá khả năng va chạm giữa các đối tượng trong quá trình chuyển động.

Định nghĩa 3.1: Đối tượng được hiểu là một tập hợp các điểm $(x, y, z) \in \Omega \in R^3$. Trong đó Ω là một miền kín giới nội.

Định nghĩa 3.2: Khoảng cách giữa 2 điểm $M_1(x_1, y_1, z_1); M_2(x_2, y_2, z_2)$ được định nghĩa là $\rho(M_1; M_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$

Định nghĩa 3.3: Xét một đối tượng được giới hạn bởi miền Ω . Khi đó đường kính của đối tượng được định nghĩa là $2r = \text{Max } \rho(M_1; M_2); \forall M_1, M_2 \in \Omega$.

Xuất phát từ **định nghĩa 3.3**, một đối tượng bất kì trong không gian R^3 luôn được mô hình hóa thành một hình cầu bán kính r với tâm chính là trọng tâm của vật kí hiệu là hình cầu $O(M, r)$ với M là tâm và r là bán kính của hình cầu.

Giả sử có m đối tượng được xác định bởi $O(M_i, r_i), i=1 \dots m$ cùng chuyển động trong một không gian quan sát H giới hạn $0 \leq x \leq x \text{ max}; 0 \leq y \leq y \text{ max}; 0 \leq z \leq z \text{ max}$; với các giả thiết sau:

- 1) Các đối tượng di chuyển theo quỹ đạo khác nhau được xác định bởi các phương trình $x = x_k(t); y = y_k(t); z = z_k(t); k, = 1, 2 \dots m$ với t : biến thời gian.

2) Tốc độ di chuyển trên các quỹ đạo của các đối tượng là khác nhau (*Tốc độ di chuyển có thể không đổi hoặc thay đổi một cách ngẫu nhiên theo thời gian*)

3) Các quỹ đạo có xảy ra các điểm cắt (*là các điểm có thể xảy ra va chạm*).

Để đánh giá khả năng va chạm giữa các vật thể, ta xét bài toán khi hai đối tượng di chuyển trên hai quỹ đạo khác nhau, tại mỗi khoảng thời gian (t), mỗi vật sẽ chiếm một khoảng không gian nào đó được xác định bởi tâm và bán kính của vật. Ta thấy rằng, hai đối tượng sẽ chỉ va chạm khi hai hình cầu đó có miền giao nhau khác rỗng, do vậy hoàn toàn có thể dựa vào thể tích miền giao để đánh giá khả năng va chạm giữa hai đối tượng bất kì.

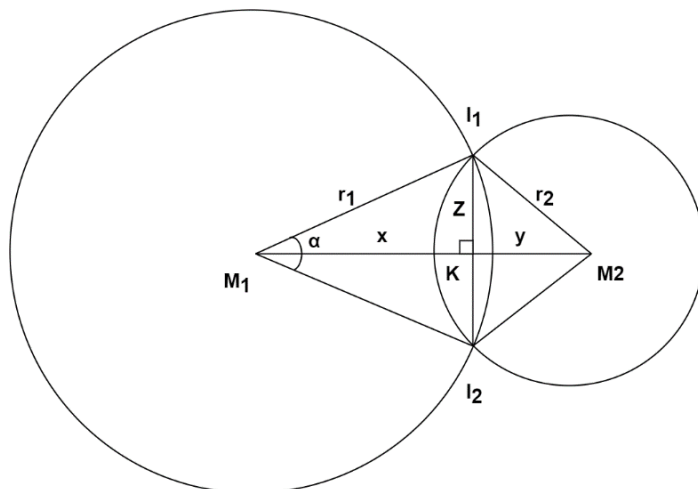
Khi thời gian (t) thay đổi cũng như vận tốc chuyển động thay đổi thì miền giao cũng thay đổi theo. Do vậy, diện tích miền giao cũng chỉ đặc trưng cho khả năng xảy ra va chạm giữa hai vật thể. Ví dụ: hai ô tô đi qua ngã tư thì điểm giao cắt có khả năng va chạm là lớn nhất, tuy nhiên trong thực tế cũng chưa chắc xảy ra va chạm nếu hai vật đi qua ngã tư, với các vận tốc khác nhau.

Xuất phát từ nhận xét trên, ta xác định xác suất xảy ra va chạm như sau: hai đối tượng được xác giới hạn bởi hai miền Ω_1, Ω_2 thì xác suất xảy ra va chạm giữa 2 đối tượng được xác định bởi công thức (3.1) với V là thể tích các miền tương ứng.

$$p[1, 2, t] = \frac{V(\Omega_1 \cap \Omega_2)}{V(\Omega_1 \cup \Omega_2)} \quad (3.1)$$

Trong trường hợp đặc biệt khi hai đối tượng di chuyển trùng khít vào nhau thì xác suất bằng 1 và khi hai đối tượng tách rời nhau thì xác suất bằng không.

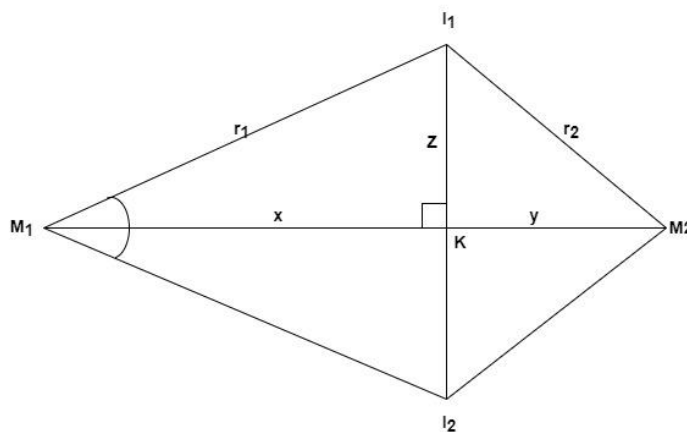
3.2.1.2. Công thức tính toán va chạm trong không gian 2 chiều



Hình 3.1. Vùng va chạm

Xét 2 đối tượng được mô tả bằng hai mặt tròn $O(M_1, r_1)$, $O(M_2, r_2)$. Kí hiệu $d = \rho(M_1, M_2)$. là khoảng cách giữa 2 tâm. Điều kiện để 2 mặt tròn giao nhau là:

$$d < r_1 + r_2.$$



Hình 3.2. Cơ sở hình học tính toán va chạm 2D

Kí hiệu $x = M_1K$; $y = M_2K$; $z = I_1K$. Khi đó chiều cao các chỏm của 2 mặt tròn tương ứng được tính theo các công thức $h_1 = r_1 - x$; $h_2 = r_2 - y$.

Ta có: $x + y = d$. (*)

Xét hai tam giác vuông (M_1KI_1) , (M_2KI_1) ta có: $r_1^2 = x^2 + z^2$; $r_2^2 = y^2 + z^2$;

Từ đó suy ra: $x^2 - y^2 = r_1^2 - r_2^2$; (**)

Từ (*) và (**) suy ra $x = \frac{r_1^2 + d^2 - r_2^2}{2d}$; $y = \frac{r_2^2 + d^2 - r_1^2}{2d}$;

Ta thu được: $h_1 = r_1 - \frac{r_1^2 + d^2 - r_2^2}{2d}$; $h_2 = r_2 - \frac{r_2^2 + d^2 - r_1^2}{2d}$

Ta có $\cos \frac{\alpha}{2} = \frac{x}{r_1} = \frac{r_1 - h_1}{r_1} \Rightarrow \alpha = 2ar \cos\left(\frac{r_1 - h_1}{r_1}\right)$ (đơn vị Radian)

Đổi giá trị góc α ra đơn vị độ, ta thu được:

$$\alpha = 2ar \cos\left(\frac{R_1 - h_1}{R_1}\right) \times \frac{180}{\pi} \text{ (đơn vị độ)}$$

Từ đó diện tích miền quạt tròn của hình tròn $O(M_1, r_1)$ được xác định bằng công

thức $S_1 = \pi r_1^2 \frac{\alpha}{360} = r_1^2 \times ar \cos\left(\frac{r_1 - h_1}{r_1}\right)$

Tương tự diện tích miền quạt tròn của hình tròn $O(M_2, r_2)$ được xác định bằng

công thức $S_2 = r_2^2 \times ar \cos\left(\frac{r_2 - h_2}{r_2}\right)$.

Để thấy rằng $S(O_1 \cap O_2) = S_1 + S_2 - S(M_1 I_1 I_2) - S(M_2 I_1 I_2)$. Vì

$$S(M_1 I_1 I_2) = x \times z = (r_1 - h_1) \times \sqrt{r_1^2 - (r_1 - h_1)^2};$$

$$S(M_2 I_1 I_2) = y \times z = (r_2 - h_2) \times \sqrt{r_2^2 - (r_2 - h_2)^2}.$$

Do đó ta thu được công thức:

$$S(O_1 \cap O_2) = r_1^2 ar \cos\left(\frac{r_1 - h_1}{r_1}\right) - (r_1 - h_1) \times \sqrt{r_1^2 - (r_1 - h_1)^2} \\ + r_2^2 ar \cos\left(\frac{r_2 - h_2}{r_2}\right) - (r_2 - h_2) \times \sqrt{r_2^2 - (r_2 - h_2)^2} \quad (3.2)$$

$$S(O_1 \cup O_2) = \pi r_1^2 + \pi r_2^2 - S(O_1 \cap O_2).$$

Trong đó $h_1 = \frac{r_1^2 + d^2 - r_2^2}{2d}$; $h_2 = \frac{r_2^2 + d^2 - r_1^2}{2d}$;

Xuất phát từ các kết quả trên, ta có xác suất va chạm giữa 2 vật được xác định bằng công thức tổng quát:

$$p[1, 2, t] = \begin{cases} \frac{S(O_1 \cap O_2)}{S(O_1 \cup O_2)}; & |r_1 - r_2| \leq d < r_1 + r_2; \\ 0; & d \geq r_1 + r_2; \\ \frac{\min(\pi r_1^2, \pi r_2^2)}{\max(\pi r_1^2, \pi r_2^2)}; & d < |r_1 - r_2|. \end{cases} \quad (3.3)$$

3.2.1.3. Công thức tính toán va chạm trong không gian 3 chiều

Xét hai đối tượng được mô tả bằng hai hình cầu $O(M_1, r_1)$, $O(M_2, r_2)$. Kí hiệu $d = \rho(M_1, M_2)$ là khoảng cách giữa 2 tâm. Điều kiện để 2 hình cầu giao nhau là:

$$d < r_1 + r_2. \quad (3.4)$$

Hoàn toàn tương tự, ta xác định được công thức tính chiều cao các chỏm cầu

$$h_1 = r_1 - \frac{r_1^2 + d^2 - r_2^2}{2d}; h_2 = r_2 - \frac{r_2^2 + d^2 - r_1^2}{2d};$$

Thể tích miền giao giữa hai hình cầu được xác định bởi công thức

$$\begin{aligned} V(O_1 \cap O_2) &= \pi h_1^2 \left(r_1 - \frac{h_1}{3}\right) + \pi h_2^2 \left(r_2 - \frac{h_2}{3}\right). \\ V(O_1 \cup O_2) &= \frac{4}{3} \pi (r_1^3 + r_2^3) - V(O_1 \cap O_2). \end{aligned} \quad (3.5)$$

Xuất phát từ các kết quả trên, ta có xác suất va chạm giữa 2 vật được xác định bằng công thức tổng quát:

$$p[1, 2, t] = \begin{cases} \frac{\pi h_1^2 \left(r_1 - \frac{h_1}{3}\right) + \pi h_2^2 \left(r_2 - \frac{h_2}{3}\right)}{\frac{4}{3} \pi (r_1^3 + r_2^3) - \pi h_1^2 \left(r_1 - \frac{h_1}{3}\right) - \pi h_2^2 \left(r_2 - \frac{h_2}{3}\right)}; & |r_1 - r_2| \leq d < r_1 + r_2; \\ 0; & d \geq r_1 + r_2; \\ \frac{\min\left(\frac{4}{3} \pi r_1^3, \frac{4}{3} \pi r_2^3\right)}{\max\left(\frac{4}{3} \pi r_1^3, \frac{4}{3} \pi r_2^3\right)}; & d < |r_1 - r_2|. \end{cases} \quad (3.6)$$

Trong trường hợp tổng quát, xét m đối tượng được xác giới hạn bởi các miền $\Omega_i, i = 1, 2, \dots, m$ thì xác suất va chạm giữa m đối tượng tại thời điểm (t) được xác định bởi công thức (3.7) trong đó V là thể tích các miền tương ứng.

$$p[1, 2, \dots, m, t] = \frac{\sum_{i \neq j}^m V(\Omega_i \cap \Omega_j)}{\sum_{i \neq j}^m V(\Omega_i \cup \Omega_j)}. \quad (3.7)$$

3.2.2. Miền đồng mức xác suất va chạm

3.2.2.1. Các định nghĩa

Xét hai đối tượng được giới hạn bởi hai miền Ω_1, Ω_2 . Giả sử tại thời điểm (t) , hai vật va chạm với xác suất $p[1, 2, t]$. Khi đó, tập hợp tất cả các điểm thuộc hai đối tượng cũng chịu sự va chạm. Như vậy, giá trị $p[1, 2, t]$ được sử dụng để đánh giá tỉ lệ va chạm tại một điểm $M(x, y, z)$ bất kì.

Định nghĩa 3.4: Xác suất va chạm tại 1 điểm thuộc một trong hai đối tượng bằng xác suất va chạm giữa hai đối tượng tại thời điểm t .

$$p(M, t) = \begin{cases} p[1, 2, t], & M \in O(M_1, R_1) \cup O(M_2, R_2); \\ 0 & M \notin O(M_1, R_1) \cup O(M_2, R_2). \end{cases} \quad (3.8)$$

Khi thời gian (t) thay đổi, xác suất va chạm tại điểm $M(x, y, z)$ cũng sẽ thay đổi và khi đó xác suất va chạm tại điểm M sẽ được hiểu là xác suất lớn nhất đạt được tại mọi thời điểm.

Định nghĩa 3.5: Xác suất va chạm tại điểm $M(x, y, z)$ kí hiệu là $p(M)$ được xác định bằng công thức: $p(M) = \text{Max } p(M, t); \forall t \in [0, T]$.

Xét bài toán nghiên cứu chuyển động của hai đối tượng $O(M_1, R_1)$ và $O(M_2, R_2)$ trong miền cửa sổ quan sát H . Ta chia miền $[0, 1]$ thành K đoạn với bước lưới $h = \frac{1}{K}$ bởi các điểm chia

$$0 = p_0 < p_1 < \dots < p_K = 1; \quad p_i = ih, i = 0, \dots, K.$$

Theo định nghĩa 3.5, mỗi điểm M nằm trong vùng quan sát H sẽ tương ứng với 1 giá trị xác suất va chạm $p(M)$. khi đó ta có định nghĩa

Định nghĩa 3.6: Tập hợp tất cả các điểm $M(x,y,z)$ thỏa mãn điều kiện $p_i < p(M(x,y,z) \leq p_{i+1}$ sẽ lập nên một miền đồng mức xác suất va chạm và được kí hiệu là miền đồng mức S_{i+1} ($i=0,1,2,\dots,K-1$).

Tập tất cả các điểm thuộc cửa sổ quan sát H luôn được chia thành các miền đồng mức xác suất va chạm S_i , ($i=0,\dots,K$) (Các điểm có xác suất va chạm $p(M)=0$ được coi là thuộc miền đồng mức 0).

Vấn đề quan trọng là xác định phương pháp để chia miền cửa sổ quan sát H thành các miền đồng mức xác suất va chạm, từ đó làm cơ sở để xây dựng thuật toán xác định quỹ đạo chuyển động cho robot đi xuyên qua các miền đồng mức với xác suất thấp để tối ưu hóa đường đi trong bài toán điều khiển robot.

3.2.2.2. Thuật toán xác định các miền đồng mức

Xuất phát từ các định nghĩa về xác suất va chạm và miền đồng mức xác suất va chạm, chúng ta nghiên cứu xây dựng thuật toán xác định tất cả các miền đồng mức trong trường hợp có hai đối tượng chuyển động trong miền quan sát H.

Tư tưởng của thuật toán

1. Chia miền H thành các lưới điểm lập nên một không gian lưới

$$H_h = (x_i, y_j, z_k); \quad x_i = ih_x; \quad y_j = jh_y; \quad z_k = kh_z;$$

$$h_x = \frac{x \max}{N}; \quad h_y = \frac{y \max}{N}; \quad h_z = \frac{z \max}{N};$$

2. Tại mỗi thời điểm t, cho hai đối tượng chuyển động và tính tọa độ các điểm tâm của vật $M_1(x_1, y_1, z_1); M_2(x_2, y_2, z_2)$. Việc tính các giá trị tọa độ dựa theo phương trình tham số của quỹ đạo chuyển động và véc tơ vận tốc chuyển động của các vật.
3. Tính khoảng cách giữa hai tâm của hai vật thể: $\rho(M_1, M_2)$
4. Nếu $\rho(M_1, M_2) < R_1 + R_2$ tức là thỏa mãn điều kiện xảy ra va chạm thì tính xác suất va chạm $p(1,2,t)$ theo công thức (3.3).
5. Duyệt mọi điểm $M(X_i, Y_i, Z_i)$ thuộc không gian lưới H_h , nếu điểm lưới thuộc một trong hai hình cầu tức là thỏa mãn

$$(X_i - x_1)^2 + (Y_i - y_1)^2 + (Z_i - z_1)^2 \leq R_1^2; \text{ hoặc}$$

$$(X_i - x_2)^2 + (Y_i - y_2)^2 + (Z_i - z_2)^2 \leq R_2^2;$$

6. Xác định xác suất tại điểm lưới theo giá trị $p(1,2,t)$, tại mọi thời điểm t luôn lưu lại giá trị lớn nhất.

7. Tiến hành phân miền đồng mức xác suất va chạm đối với mọi điểm lưới thuộc không gian lưới H_h .

Thuật toán phân miền đồng mức xác suất va chạm:

Input:

- $x_{max}, y_{max}, z_{max}, N$: Kích thước miền quan sát H và số điểm chia lưới.
- $x_1(t), y_1(t), z_1(t), (u_1, v_1, w_1)$: Quỹ đạo, véc tơ vận tốc đối tượng 1
- $x_2(t), y_2(t), z_2(t), (u_2, v_2, w_2)$: Quỹ đạo, vectơ vận tốc đối tượng 2.
- R_1, R_2 : Bán kính 2 vật thể.
- T : Khoảng thời gian chuyển động của các vật thể.
- K : Số chia thành các miền đồng mức.

Output: Các miền đồng mức S_i ($i=0..K$).

a) Thuật toán tính xác suất va chạm

<p>Khởi tạo dữ liệu</p>	<p><i>Phân lớp xác suất va chạm: $P = \text{linspace}(0, 1, K+1)$;</i></p> <p><i>Xây dựng không gian lưới, kí hiệu $H(i, j, k)$ là các điểm lưới ($i, j, k = 0:N$)</i></p> <p>$X = \text{linspace}(0, x_{max}, N+1)$;</p> <p>$Y = \text{linspace}(0, y_{max}, N+1)$;</p> <p>$Z = \text{linspace}(0, z_{max}, N+1)$;</p> <p><i>Khởi tạo xác suất tại mọi điểm lưới: $P(X, Y, Z) = 0$; $t = 0$; $\Delta t = \frac{T}{M}$;</i></p>
<p>Thực hiện tính toán xác suất va chạm</p>	<p><i>While $t < T$</i></p> <p><i>Begin</i></p> <p>$t = t + \Delta t$;</p> <p><i>Xác định tọa độ tâm $M_1(X_1, Y_1, Z_1)$ và $M_2(X_2, Y_2, Z_2)$</i></p> <p>$X_1 = x_1(t)$; $Y_1 = y_1(t)$; $Z_1 = z_1(t)$; $X_2 = x_2(t)$; $Y_2 = y_2(t)$; $Z_2 = z_2(t)$;</p> <p><i>{Kiểm tra điều kiện va chạm}</i></p> <p><i>Tính $d = \rho(M_1, M_2)$</i></p> <p><i>if ($d < R_1 + R_2$)</i></p> <p><i>Begin</i></p> <p><i>Tính xác suất va chạm p [1, 2, t] theo công thức (3.3) hoặc (3.6)</i></p>

	<p>Duyệt mọi điểm lưới</p> <p>for $i,j,k=0:N$</p> <p>{Điểm lưới thuộc hợp của 2 hình cầu}</p> <p>if $H(i, j, k) \in O(M_1, R_1) \cup O(M_2, R_2)$ then</p> <p> if $p(1,2,t) > P(i,j,k)$ theo $P(i,j,k)=p(1,2,t)$</p> <p>End;</p> <p>End;</p>
Kết quả	Mảng dữ liệu $P(X,Y,Z)$ là xác suất va chạm của hai đối tượng

b) Thuật toán phân vùng đồng mức xác suất va chạm

Khởi tạo dữ liệu	Khởi động mức 0 cho tất cả các điểm lưới $V(i,j,k)=0$
Xác định mức va chạm cho các điểm lưới	<p>Duyệt tất cả các điểm lưới for $i,j,k=0:N$</p> <p>Đối với mỗi điểm lưới, duyệt lần lượt các mức, nếu xác suất va chạm tại điểm $H(i,j,k)$ thuộc khoảng l thì gán mức l cho $H(i,j,k)$, $l=1..K$.</p> <p>For $l=1:K$</p> <p>if $(p(H(i, j, k) > p(l)) \text{ and } (p(H(i, j, k) \leq p(l+1)))$ then $V(i, j, k) = l$;</p>
Phân vùng đồng mức va chạm	<p>For $l=0:K$</p> <p>Begin</p> <p>$S(l) = \emptyset$;</p> <p>For $i,j,k=1:N$</p> <p>if $V(i,j,k)=l$ then $S(l) = S(l) \cup \{H(i, j, k)\}$;</p> <p>end.</p>
Kết quả	Tập hợp tất cả các điểm lưới $H(i,j,k)$ có cùng một mức i lập nên tập hợp điểm chính là miền đồng mức S_i là miền có xác suất va chạm

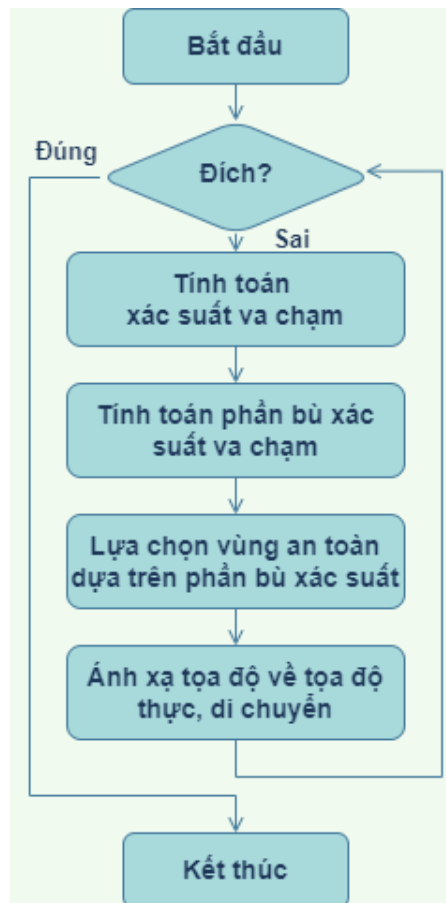
Trong đó, khoảng giá trị của $H(i,j,k)$:

$$p_l < p(H(i, j, k) \leq p_{l+1}); p_l = \frac{1}{l}. \quad (3.9)$$

Độ phức tạp thuật toán phụ thuộc vào số lần lặp theo thời gian (t) và số lần lặp trong không gian các điểm lưới. Để thấy rằng độ phức tạp của thuật toán trên tương đương với $O(MN^3)$ trong đó M là số điểm lưới thời gian và N là số điểm lưới theo chiều không gian.

3.2.3. Thuật toán tránh va chạm dựa trên xác suất va chạm

Ý tưởng phương pháp tránh va chạm cho robot tự hành được đề xuất dựa trên phân bù của xác suất va chạm tính được.



Hình 3.3. Sơ đồ thuật toán tránh va chạm dựa trên xác suất va chạm

Sau khi tính toán được xác suất va chạm, việc xây dựng hướng di chuyển, quỹ đạo di chuyển của robot tự hành sẽ được ánh xạ lên hệ thống định vị GPS trên cơ sở quỹ đạo tính toán của phân bù xác suất va chạm. Tại mỗi thời điểm (t) robot tự hành

sẽ thực hiện: 1- Thực hiện tính toán xác suất va chạm; 2- Xác định phần bù của xác suất va chạm; 3- Xây dựng đồ thị của phần bù, lựa chọn điểm di chuyển tiếp theo; 4- Thực thi chuyển động;

3.3. Kết quả tính toán và mô phỏng

3.3.1. Kết quả xác suất tránh va chạm P

Giả thiết tại mỗi thời điểm (t), ta xác nhận 02 đối tượng để tính toán xác suất va chạm và xây dựng miền đồng mức xác suất va chạm đó.

Dữ liệu đầu vào:

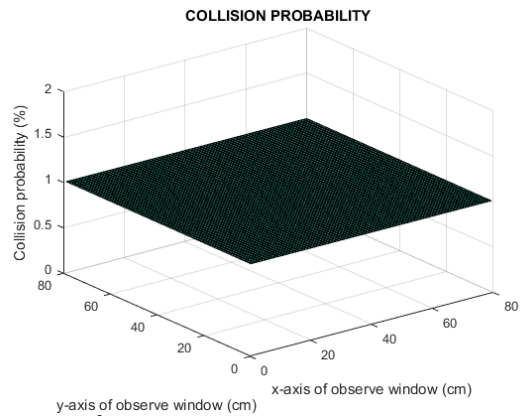
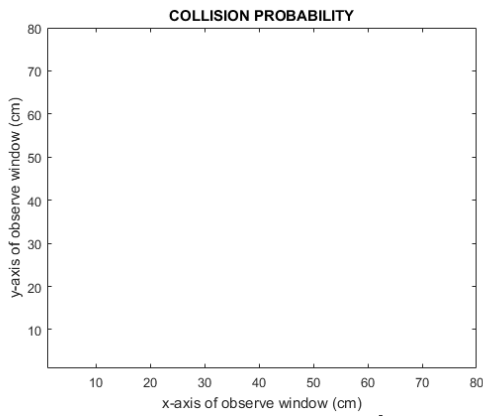
- $x_{max}, y_{max}, z_{max}$: Kích thước miền quan sát H
- N: số điểm chia lưới.
- $x_1(t), y_1(t), z_1(t), (u_1, v_1, w_1)$: Quỹ đạo, véc tơ vận tốc đối tượng 1
- $x_2(t), y_2(t), z_2(t), (u_2, v_2, w_2)$: Quỹ đạo, véc tơ vận tốc đối tượng 2.
- R_1, R_2 : Bán kính 2 vật thể.
- T: Khoảng thời gian chuyển động của các vật thể.
- K: Số chia thành các miền đồng mức.

Kết quả đầu ra: Các miền đồng mức S_i ($i=0..K$).

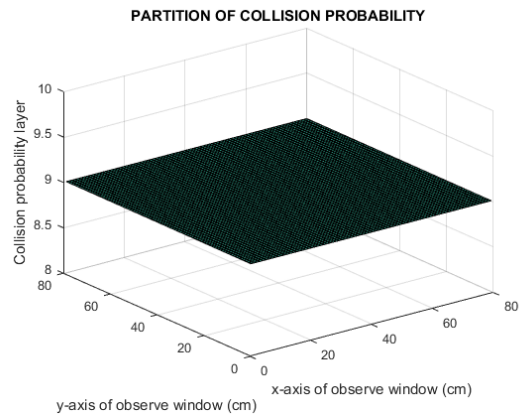
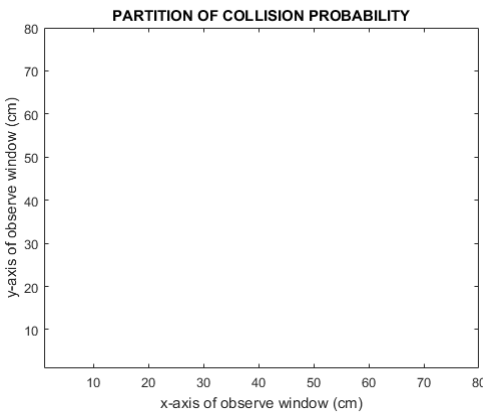
Bảng 3.1. Bảng tham số tính toán xác suất va chạm.

Trường hợp	K	R1	R2	T	N
1	10	50	30	80	80
2	10	30	30	50	50
3	5,20	50	60	100	100
4	5,10,15,20	50	60	100	100

a) Trường hợp 1



(a) Kết quả tính toán xác suất va chạm

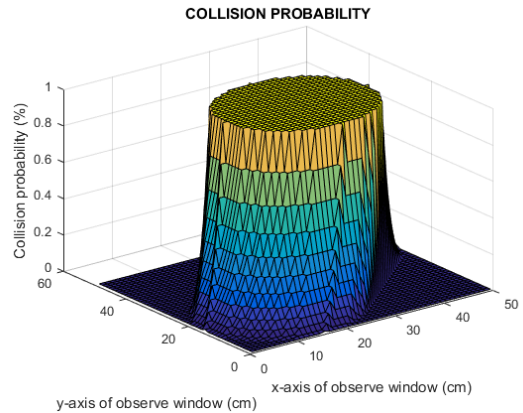
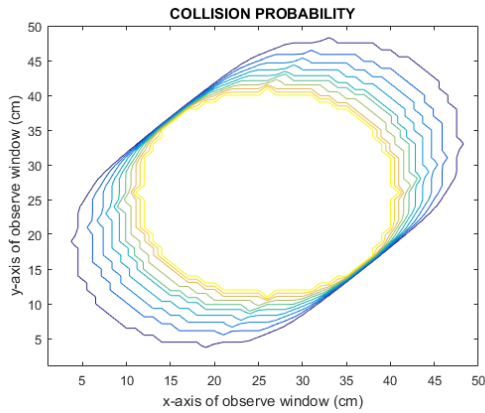


(b) Kết quả tính toán và phân vùng đồng mức xác suất va chạm

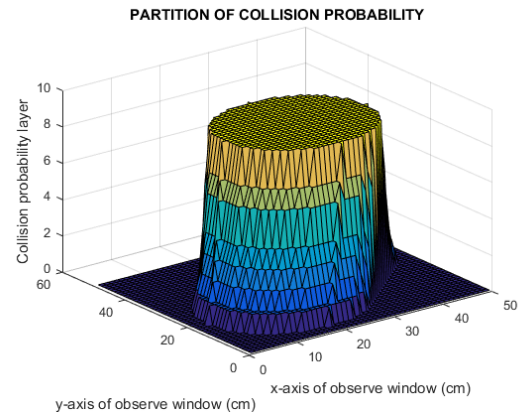
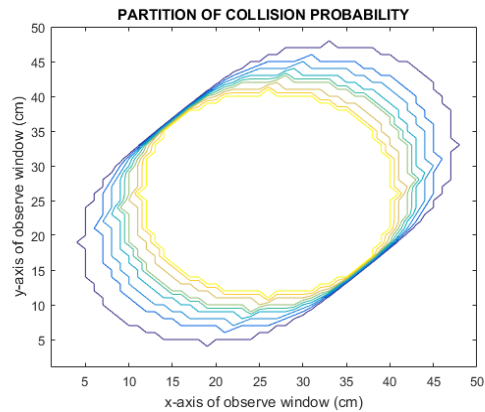
Hình 3.4. Kết quả trường hợp 1

Kết quả tính toán và mô phỏng của trường hợp 1 được mô tả trong hình 3.4. Ở đây, hai đối tượng di chuyển trùng khít vào nhau thì xác suất bằng 1, bởi vậy so sánh kết quả tính toán xác suất va chạm (collision probability) trên hình 3.4a và kết quả tính toán và phân vùng đồng mức xác suất va chạm (partition collision probability) trên hình 3.4b ta có kết quả mô phỏng giống nhau.

b) Trường hợp 2



(a) Kết quả tính toán xác suất va chạm



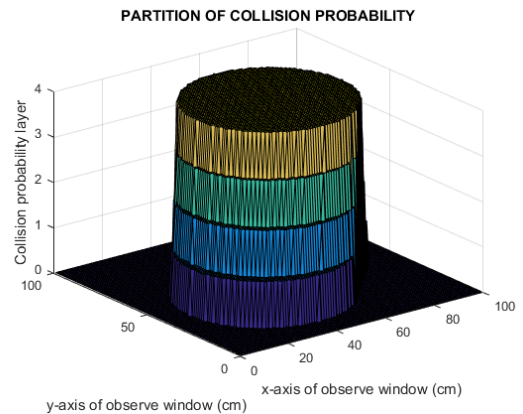
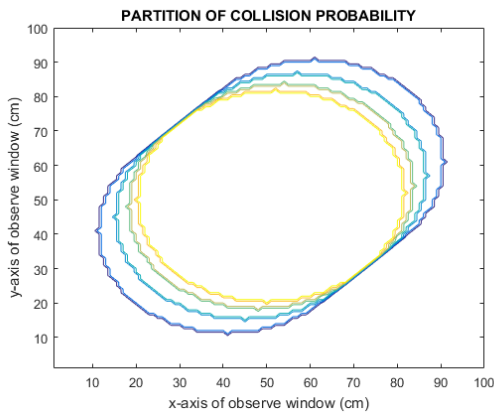
(b) Kết quả tính toán và phân vùng đồng mức xác suất va chạm

Hình 3.5. Kết quả trường hợp 2

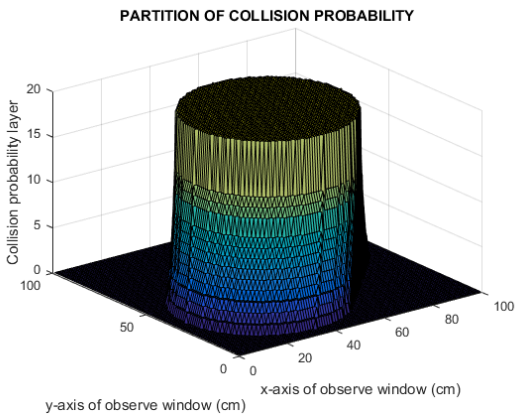
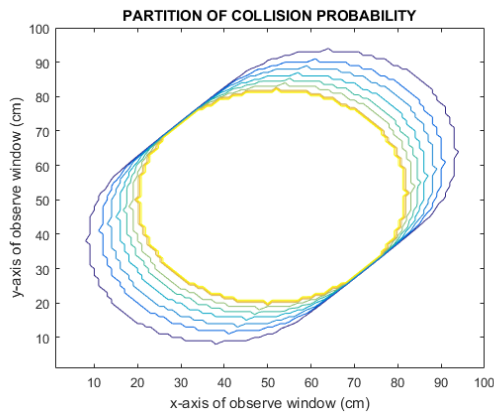
Với bộ dữ liệu của trường hợp 2, kết quả tính toán xác suất va chạm và phân vùng đồng mức xác suất va chạm được thể hiện rõ rệt khi so sánh với kết quả tính toán xác suất va chạm khi chưa được phân vùng đồng mức xác suất va chạm (Hình 3.5a) với các kết quả khi đã được phân vùng với giá trị $k=10$ (Hình 3.5b).

Ở đây, kết quả phân xác suất va chạm đã được tính toán nhằm kiểm chứng mô hình toán học xác suất, thuật toán phân vùng đồng mức xác suất va chạm cho thấy việc phân chia $k=10$ thành 10 vùng đồng mức khác nhau một cách rõ rệt, là cơ sở để robot lựa chọn, đánh giá vùng xác suất xuyên để tránh được vật cản.

c) Trường hợp 3



(a) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=5$

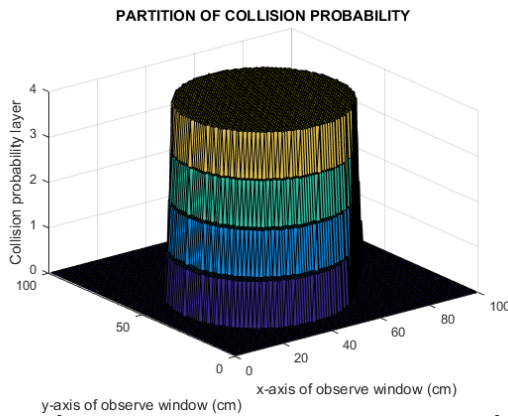


(b) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=10$

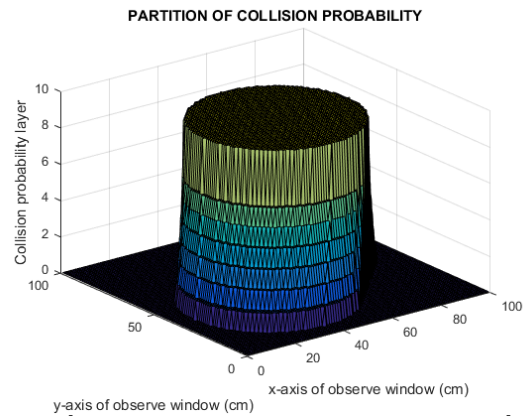
Hình 3.6. Kết quả trường hợp 3

Ở trường hợp 3, việc tính toán xác suất va chạm và phân vùng va chạm được thực hiện với cùng bộ dữ liệu đầu vào, đồng thời thay đổi số lượng phân vùng đồng mức xác suất va chạm với $k=5$ và $k=20$. Kết quả cho thấy kết quả tính toán xác suất va chạm của cả hai giá trị của k là như nhau. Tuy nhiên, kết quả tính toán và phân vùng đồng mức xác suất va chạm là khác nhau. Cụ thể: với $k=5$, phân vùng đồng mức xác suất va chạm được thể hiện trong hình 3.6a; với $k=10$, phân vùng đồng mức xác suất va chạm thể hiện trong hình 3.6b. Việc phân vùng đồng mức xác suất va chạm được chi rõ và khi số lượng phân vùng càng nhỏ thì trong các điểm nút đầu cuối của một vùng sẽ dày hơn, điều đó có nghĩa là việc ra quyết định cho robot lựa chọn vùng xuyên qua sẽ càng an toàn khi ta phân vùng thành nhiều lớp hơn.

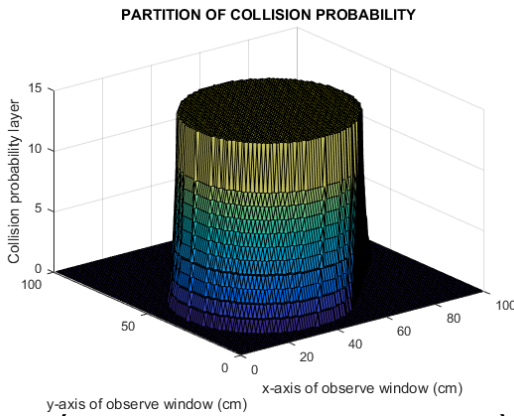
d) Trường hợp 4



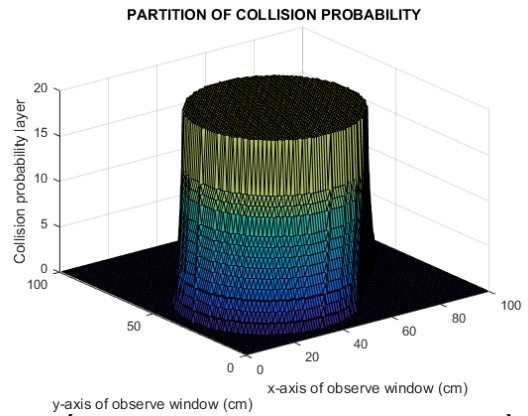
(a) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=5$



(b) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=10$



(c) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=15$



(d) Kết quả tính toán và phân vùng đồng mức xác suất va chạm với $k=20$

Hình 3.7. Kết quả trường hợp 4

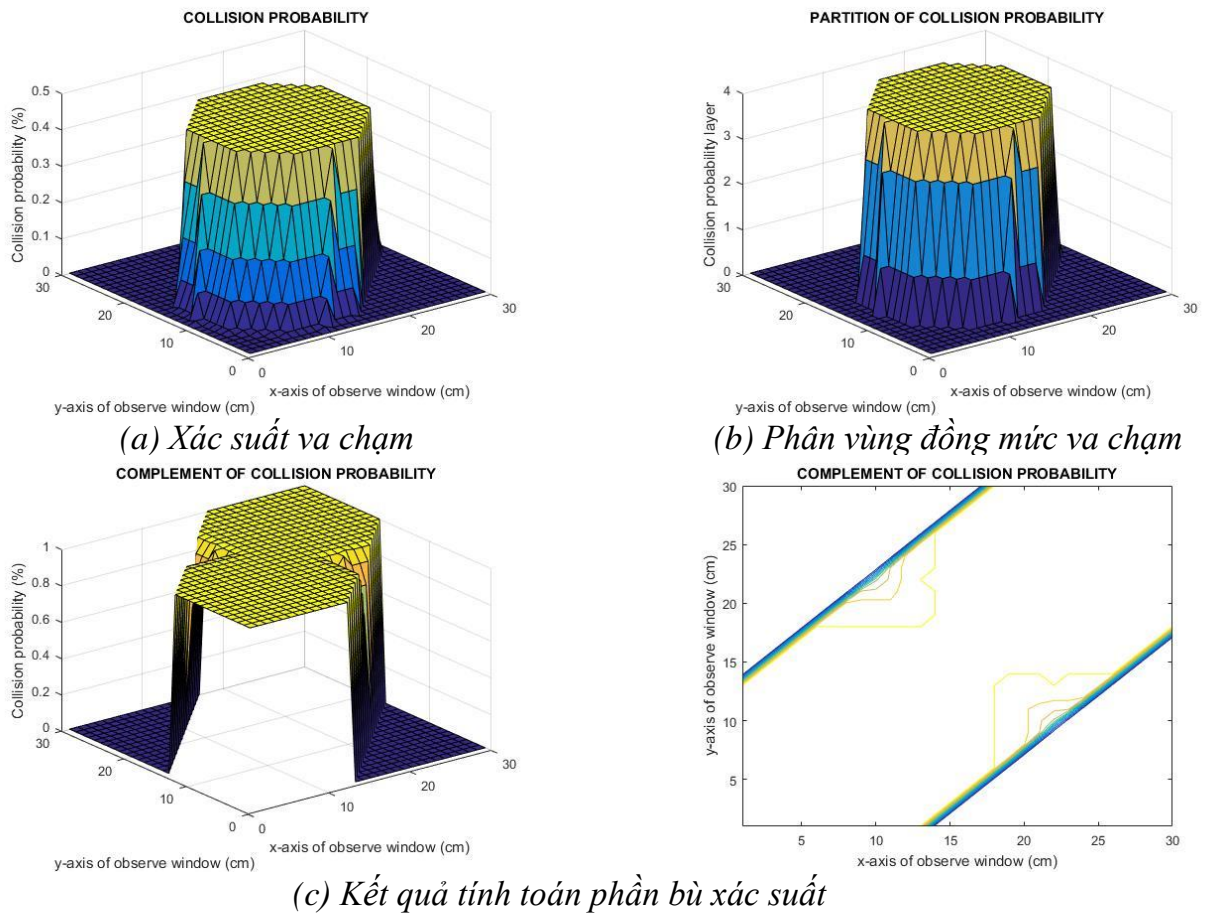
Trường hợp 4 là một phần đóng góp cho thuật toán tránh va chạm cho robot dựa trên xác suất va chạm và việc phân vùng đồng mức xác suất va chạm. Với cùng bộ dữ liệu, tiến hành kiểm thử với số lượng phân vùng khác nhau: $k=5$, $k=10$, $k=15$, $k=20$. Kết quả tính toán xác suất va chạm, phân vùng đồng mức xác suất đã được thực hiện và thể hiện trên hình 3.7.

Nhận xét: khi số lượng lớp phân vùng càng ít thì tốc độ tính toán nhanh hơn, tuy nhiên tính chính xác thấp, rủi ro cao trong việc ra quyết định di chuyển của robot. Ngược lại, nếu số lượng phân vùng xác suất va chạm được chi nhỏ, chia thành nhiều vùng hơn thì robot sẽ có thêm nhiều sự lựa chọn an toàn cho việc di chuyển và tốc độ tính toán sẽ chậm hơn.

3.3.2. Kết quả xác suất tránh va chạm (1-P)

Bảng 3.2. Dữ liệu tính toán xác suất phân bù va chạm (1-P)					
Trường hợp	K	R1	R2	T	N
1	5	20	30	30	30
2	10	50	60	100	100

a) Trường hợp 1

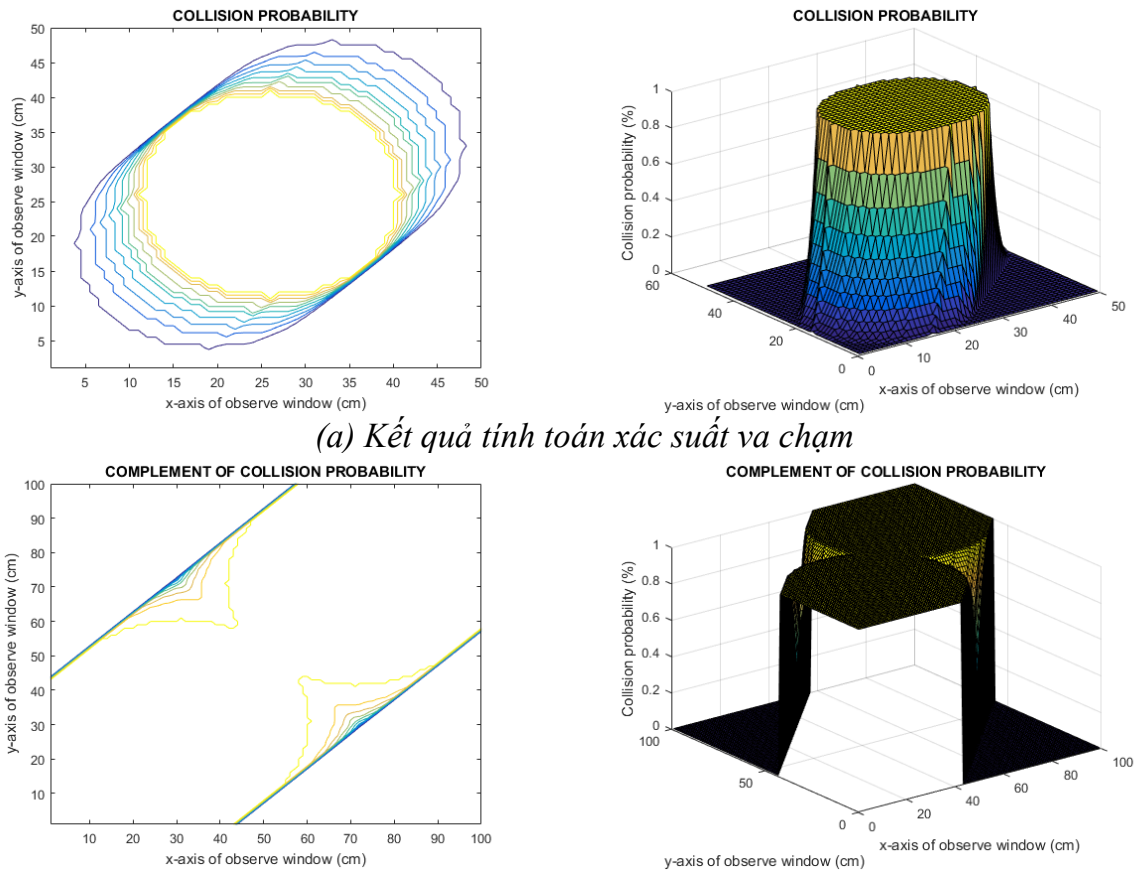


Hình 3.8. Kết quả trường hợp 1 với 1-P

Kết quả tính toán và mô phỏng ở trường hợp 1 gồm: xác suất va chạm (Hình 3.8a), phân vùng đồng mức va chạm (Hình 3.8b), tính toán phân bù xác suất 1-P (complement of collision probability) xác suất va chạm (Hình 3.8c). Đối sánh các kết quả mô phỏng, ta thấy việc tính toán xác suất va chạm trong trường hợp này đã được thực hiện, việc phân vùng xác suất theo số lớp $k=5$ cũng đã được phân tách rõ rệt.

Đồng thời, phân bù xác suất va chạm 1-P cũng đã được thể hiện rõ rệt khi đối sánh trên dữ liệu tính toán. Qua đó thể hiện sự đúng đắn của thuật toán tính toán xác suất va chạm, thuật toán phân vùng đồng mức va chạm, cũng như chỉ rõ khu vực an toàn để robot lựa chọn ra quyết định trong phân biểu đồ thể hiện phân bù xác suất va chạm.

b) Trường hợp 2



(a) Kết quả tính toán xác suất va chạm

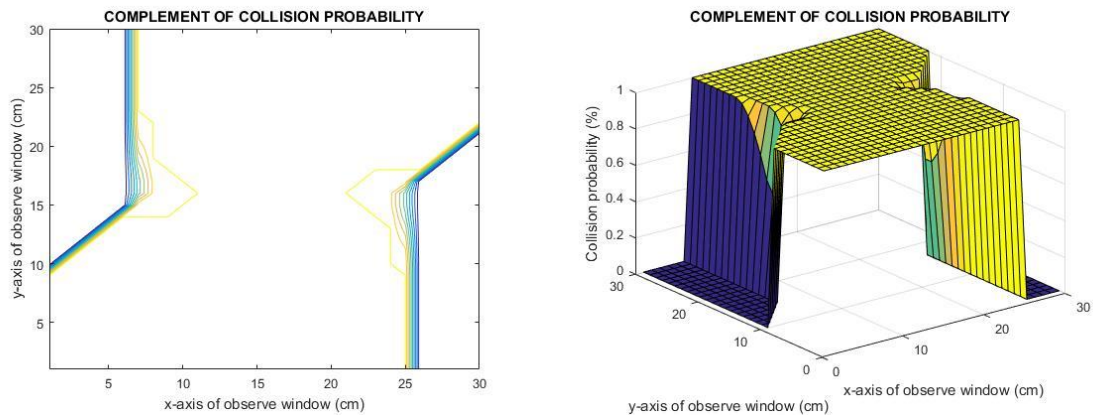
(b) Kết quả tính toán phân bù xác suất

Hình 3.9. Kết quả trường hợp 2 với 1-P

Với bộ dữ liệu của trường hợp 2, kết quả tính toán và được mô phỏng tại Hình 3.9. Kết quả hiển thị cho thấy việc tính toán xác suất và phân bù xác suất của thuật toán và công thức được trình bày ở mục 3.2.

3.3.3. Mô hình (1-P) với nhiều vật cản

Kiểm thử mô hình bù xác suất (1-P) để xây dựng quỹ đạo và phân vùng an toàn cho robot lựa chọn và di chuyển với nhiều vật cản. Áp dụng nguyên lý chồng chất đối với xác suất va chạm của robot với hai vật cản được mô tả trên Hình 3.10, ta nhận thấy xác suất va chạm tổng quát có sự giao thoa vùng xác suất thành phần. Cụ thể trong trường hợp này, sau khi tính toán xác suất va chạm của robot với từng vật cản, robot có một quỹ đạo di chuyển an toàn và rõ ràng. Điều này cho thấy phương pháp tính toán xác suất va chạm, phân vùng va chạm cũng như thuật toán xác định đường đi, tránh va chạm dựa trên phần bù xác suất va chạm (1-P) là hoàn toàn có cơ sở và khẳng định tính đúng đắn của phương pháp.



Hình 3.10. Kết quả tính toán và mô phỏng (1-P) cho nhiều vật cản

3.2.4. Nhận xét

Phương pháp tối ưu hóa đường đi cho robot dựa trên xác suất va chạm đã được trình bày là cơ sở để robot ra quyết định di chuyển qua vùng an toàn và xây dựng quỹ đạo cho nó. Kết quả tính toán đã kiểm chứng cơ sở toán học được trình bày.

Việc mở rộng hướng nghiên cứu lớp bài toán tránh va chạm với các vật cản chuyển động, có kể đến yếu tố môi trường sẽ tiếp tục được thực hiện.

Việc cải tiến thuật toán phân vùng đồng mức va chạm tiếp tục thực hiện nhằm hỗ trợ robot đưa ra các quyết định chính xác, an toàn và tin cậy trong bài toán dự báo, phát hiện va chạm.

3.4. Kết luận và vấn đề nghiên cứu

Chương 3 của luận án đề xuất một phương pháp mới để tính toán va chạm dựa trên xác suất trên cơ sở toán học thể tích. Các mô hình xác suất P và bù xác suất $(1-P)$ và thuật toán phân vùng đồng mức xác suất va chạm là cơ sở đề xuất thuật toán phát hiện, tránh va chạm cho robot bằng việc lựa chọn vùng an toàn, tránh va chạm trong quá trình di chuyển. Kết quả mô phỏng đã kiểm chứng mô hình toán học được nghiên cứu, là cơ sở cho việc xây dựng, cài đặt thuật toán phát hiện, tránh va chạm cho robot.

PHẦN KẾT LUẬN

Kho hàng thông minh được xem là một giải pháp hiệu quả để lưu trữ hàng hóa, hơn hẳn những phương pháp lưu trữ kho truyền thống. Các hệ thống này phù hợp với những nhà máy, doanh nghiệp sản xuất hay kinh doanh trong nhiều lĩnh vực như: các ngành sản xuất, chế biến thực phẩm, dược phẩm, hàng tiêu dùng, linh kiện điện tử... Những doanh nghiệp lớn trên thế giới như Amazon, Walmart, Alibaba, DHL... hay Vinamilk ở Việt Nam đã và đang ứng dụng có hiệu quả kho hàng thông minh trong việc lưu trữ và quản lý hàng hóa. Đây là một xu hướng cho các doanh nghiệp dịch chuyển từ kho truyền thống với lượng lớn nhân công và máy móc hỗ trợ sang kho hàng thông minh nhằm nâng cao năng lực cạnh tranh, giảm chi phí.

Hiện nay, kho hàng thông minh đang là xu thế phát triển toàn cầu, tiêu biểu của công nghệ 4.0 với sự tham gia, kết nối các công nghệ tiên tiến trong quy trình kho và quản lý hoạt động kho. Ngoài việc sử dụng các thiết bị tự động hóa như cảm biến, máy móc để tự động hóa các hoạt động trong kho hàng thì kho hàng thông minh hiện nay đã ứng dụng nhiều hơn công nghệ robot, IoT và trí tuệ nhân tạo để thực hiện các nhiệm vụ quản lý hàng tồn kho và định vị sản phẩm xung quanh kho thay thế lao động thủ công. Không thể phủ nhận được vai trò trụ cột của các robot trong công đoạn tự động hóa kho hàng, thực hiện quy trình quản lý, vận hành kho hiệu quả, trong đó robot tự hành là chủ yếu thực thi các công việc vận tải, sắp xếp, kiểm đếm hàng hóa trong kho hàng thông minh.

Vậy nhưng, một thách thức lớn mà các robot tự hành đang phải đối diện và cần giải quyết đó là nguy cơ xảy ra va chạm trong quá trình làm việc với các thành phần khác trong cùng môi trường hoạt động. Bởi vậy, nhiều nhà sản xuất, nhiều nhà nghiên cứu đã và đang quan tâm đến vấn đề tránh va chạm cho robot tự hành nhằm hướng đến tính đa dạng hơn trong ứng dụng, tăng tính hiệu quả và các robot này trong các lĩnh vực hoạt động bao gồm cả trong quản lý kho hàng thông minh.

Bởi vậy, luận án này đã tập trung nghiên cứu giải quyết vấn đề tránh va chạm cho robot tự hành theo hướng tiếp cận tránh va chạm dựa trên phản ứng, góp một

phần giải quyết thách thức đang đặt ra với hệ thống các robot tự hành. Luận án gồm 5 phần, với 3 chương nội dung:

1. Tổng quan về robot tự hành, ứng dụng, thách thức và vấn đề tránh va chạm.

2. Nghiên cứu và phát triển kĩ thuật tránh va chạm cho robot tự hành dựa trên cấu trúc cây phân lớp hệ bao (BVH), kết hợp với thuật toán Elastic strips nhằm nâng cao hiệu quả phát hiện, tránh va chạm.

3. Nghiên cứu và đề xuất kĩ thuật phát hiện, tránh va chạm cho robot tự hành dựa trên mô hình tính toán xác suất va chạm, phân vùng đồng mức xác suất va chạm.

Đồng thời, tác giả cũng đã đề xuất các mô hình tính toán xác suất va chạm dựa trên mô hình toán học thể tích; thuật toán phân vùng đồng mức va chạm, mô hình xác suất va chạm (P) và phân bù xác suất va chạm (1-P) của các đối tượng;

Kết quả chính của luận án:

Thứ nhất, đề xuất thuật toán phát hiện, tránh va chạm cho robot tự hành dựa trên cấu trúc cây phân lớp hệ bao BVH, kết hợp với thuật toán Elastic strips nhằm nâng cao hiệu quả của việc phát hiện và tránh va chạm.

Thứ hai, đề xuất mô hình tính toán xác suất va chạm dựa trên mô hình toán học thể tích; thuật toán phân vùng đồng mức va chạm, mô hình xác suất va chạm (P) và phân bù xác suất va chạm (1-P) của các đối tượng và thuật toán phát hiện, tránh va chạm cho robot tự hành dựa trên phân vùng đồng mức va chạm.

Việc cài đặt và kiểm thử các thuật toán với mô hình robot thực tế sẽ được thực hiện và không trình bày trong luận án.

Hướng phát triển của luận án:

Thứ nhất, tiếp tục mở rộng nghiên cứu mô hình xác suất va chạm P và (1-P) và thuật toán phát hiện, tránh va chạm cho robot tự hành trong môi trường kết hợp nhiều robot làm việc song song.

Thứ hai, việc phát triển hướng nghiên cứu sẽ được tính toán và cài đặt kiểm thử thuật toán phát hiện, tránh va chạm cho robot trong thực tế.

Thứ ba, tiếp tục nghiên cứu và phát triển, cải tiến kĩ thuật đánh giá vùng đồng mức xác suất va chạm trên lí thuyết nội suy kết hợp mô hình Markov nhằm tối ưu hóa tính toán.

DANH MỤC CÔNG BỐ

- CT1 Nông Minh Ngọc, Phạm Đình Lâm, Đỗ Năng Toàn (2016), “Thuật toán tránh va chạm cho robot dựa trên cải tiến kỹ thuật Elastic strips”, *Tạp chí Nghiên cứu KH&CN quân sự*, Số Đặc san ACMEC 07, pp.92-102.
- CT2 Nong Minh Ngọc, Do Nang Toan (2017), “A Collision Avoidance Algorithm based on Elastic strips using bounding volume (BV)”, *19th International Conference on Robot Software and Robot Programming*, 2280-2284.
- CT3 Khanh Duong-Quoc, Thuy Le-Thi-Thu, Long Pham-Thanh, Ngoc Nong-Minh (2022), “Controlling Hybrid Machine Tools concerning Error Compensation of Chain Elements”, *Journal of Robotics*, Volume 2022, Article ID 4366888, 9 pages.
- CT4 Nong Minh Ngọc, Do Nang Toan, Vu Vinh Quang, Ngo Ngoc Vu (2022), “A method of obstacle avoidance for amr robot in warehouse automation”, *TNU Journal of Science and Technology*, Volume 228(02), pp.62 – 69.
- CT5 Nong Minh Ngọc, Do Nang Toan, Vu Vinh Quang, Vu Ngo Ngoc (2023), “Obstacle avoidance algorithm based on the probability of collision”, *International Journal of Engineering Trends and Technology*, Volume 71(3), pp.328-336.

TÀI LIỆU THAM KHẢO

1. Abed, Mustafa, Lutfy, Omar, and Al-Doori, Qusay (2021), "A Review on Path Planning Algorithms for Mobile Robots", *Engineering and Technology Journal*. 39(5), pp. 804-820.
2. Alatisse, Mary B., Gerhard P. Hancke (2020), "A review on challenges of autonomous mobile robot and sensor fusion methods", *IEEE Access*. 8, pp. 39830-39846.
3. Alfano, Salvatore (2004), Determining probability upper bounds for NEO close approaches, *2004 Planetary Defense Conference: Protecting Earth from Asteroids*, p. 1478.
4. Alfano, Salvatore (2005), "Relating position uncertainty to maximum conjunction probability", *The Journal of the Astronautical Sciences*. 53, pp. 193-205.
5. Alfriend, Kyle T Akella, Maruthi R Frisbee, Joseph Foster, James L Lee, Deok-Jin Wilkins, Matthew (1999), "Probability of collision error analysis", *Space Debris*. 1(1), pp. 21-35.
6. Arkin, Ronald C (1989), "Motor schema—based mobile robot navigation", *The International journal of robotics research*. 8(4), pp. 92-112.
7. Berns, Karsten, and Syed Atif Mehdi (2010), Use of an autonomous mobile robot for elderly care, *2010 Advanced Technologies for Enhancing Quality of Life*, IEEE, Iasi, Romania, pp. 121-126.
8. Borenstein, Johann and Koren, Yoram and automation (1991), "Histogramic in-motion mapping for mobile robot obstacle avoidance", *IEEE Transactions on robotics*. 7(4), pp. 535-539.
9. Borenstein, Johann, Yoram Koren (1988), "Obstacle avoidance with ultrasonic sensors", *IEEE Journal on Robotics and Automation*. 4(2), pp. 213-218.

10. Borenstein, Johann, Yoram Koren (1991), "The vector field histogram-fast obstacle avoidance for mobile robots", *IEEE transactions on robotics and automation* 7(3), pp. 278-288.
11. Boxerbaum AS, Chiel HJ, and Quinn RD (2009), "Softworm: a soft, biologically inspired worm-like robot", *Neuroscience Abstracts*. 315(5817), p. 44106.
12. Brooks, Rodney (1986), "A robust layered control system for a mobile robot.", *IEEE journal on robotics and automation* 2(1), pp. 14-23.
13. Brooks, Rodney A., Jonathan H. Connell (1987), Asynchronous distributed control system for a mobile robot, *Mobile Robots I*, SPIE, pp. 77-84.
14. Chan, F Kenneth (2008), *Spacecraft collision probability*, Aerospace Press El Segundo, CA.
15. Chen, D., Dai, C., Wan, X., Mou, J. (2015), A research on AIS-based embedded system for ship collision avoidance, *2015 International Conference on Transportation Information and Safety (ICTIS)*, IEEE, pp. 512-517.
16. Chen, Yunxia, Yaohua Wu, Han Xing (2017), "A complete solution for AGV SLAM integrated with navigation in modern warehouse environment", *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 6418-6423.
17. Chung, Hakyoun, Yong Seek Choi, Jang Gyu Lee (1992), Path planning for a mobile robot with grid type world model, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, pp. 439-444.
18. Coppola, Vincent T Woodburn, James Hujsak, Richard (2004), Effects of cross correlated covariance on spacecraft collision probability, *AAS/AIAA Spaceflight Mechanics Meeting*.
19. Corke, P., Trevelyan, J., Brock, O., Khatib, O. (2000), "Elastic strips: A framework for integrated planning and execution", *Experimental Robotics VI*, pp. 329-338.

20. Corke, Peter Roberts, Jonathan Cunningham, Jock Hainsworth, David (2008), "Mining robotics", *Springer handbook of robotics*, pp. 1127-1150.
21. Crnokic, Boris, Miroslav Grubisic, and Tomislav Volaric (2017), "Different applications of mobile robots in education", *International Journal on Integrating Technology in Education*. 6(3), pp. 15-28.
22. Di Paola, Donato, et al (2010), "An autonomous mobile robotic system for surveillance of indoor environments", *International Journal of Advanced Robotic Systems*. 7(1), pp. 19-26.
23. Dijkstra, E. W. (1959), "A note on two problems in connexion with graphs", *Numerische Mathematik*. 1(1), pp. 269–271.
24. Eberly, D. (2008), *Dynamic Collision Detection using Oriented Bounding Boxes*, accessed November 30-2022, from <http://www.geometrictools.com>
25. Eberly, David *Numerical methods for ordinary differential equations*, Technical report, Geometric Tools, Inc, accessed November 30-2022, from <http://www.geometrictools.com>.
26. Estier, Thomas, et al. (2000), Shrimp, a rover architecture for long range martian mission.", *Proceedings of the Sixth ESA Workshop on Advanced Space Technologies for Robotics and Automation ASTRA 2000*, Noordwijk, The Netherlands.
27. F. A. Raheem, U. I. Hameed (2018), "Interactive heuristic D* path planning solution based on PSO for two-link robotic arm in dynamic environment", *World J. Eng. Technol.* 7(1), pp. 80–99.
28. Frisbee Jr, Joseph H (2015), An upper bound on orbital debris collision probability when only one object has position uncertainty information, *AAS/AIAA Astrodynamics Specialist Conference*.
29. Fulgenzi, Chiara, Anne Spalanzani, Christian Laugier (2007), Combining Probabilistic Velocity Obstacles and Occupancy Grid for safe Navigation in dynamic environments, *Workshop on safe Navigation, IEEE ICRA*.

30. Fulgenzi, Chiara, Anne Spalanzani, Christian Laugier (2007), Dynamic obstacle avoidance in uncertain environment combining PVOs and occupancy grid, *Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007.*
31. Goris, Kristof (2005), *Autonomous mobile robot mechanical design*, Vrije Universiteit Brussel, Brussels.
32. Gottlieb, Robert G Sponaugle, Steven J Gaylor, David E (2001), Orbit determination accuracy requirements for collision avoidance, *Proceeding of the 11 th Annual AAS/AIAA Space Flight Mechanics Meeting*, Santa Barbara, CA, pp. 1105-1121.
33. Guo, Lixiao, Qiang Yang, Wenjun Yan (2012), Intelligent path planning for automated guided vehicles system based on topological map, *2012 IEEE conference on control, systems & industrial informatics*, IEEE, pp. 69-74.
34. Hancke, Gerhard P., Konstantinos Markantonakis, and Keith E. Mayes. (2010), "Security challenges for user-oriented RFID applications within the Internet of things", *Journal of Internet Technology*. 11(3), pp. 307-313.
35. Hussain, A. A. (2017), *New Approaches for C-Space Map Construction for Heuristic Path Solutions Based on PSO Technique*, Master Thesis, Univ. Technol.
36. Hyun, Kyung, Minh Ngoc, M. Asif Ali (2008), "A real time collision avoidance algorithm for mobile robot based on elastic force", *International Journal of Mechanical and Mechatronics Engineering* 2(5), pp. 666-669.
37. Ishihara, S., Kanai, M., Narikawa, R., Ohtsuka, T. (2022), "A Proposal of Path Planning for Robots in Warehouses by Model Predictive Control without Using Global Paths", *IFAC-PapersOnLine*. 55(37), pp. 573-578.
38. J, Paul (2013), *Military robots and drones: a reference handbook*, ABC-CLIO, Springer.
39. Kamamichi N, Yamakita M, Asaka K, et al (2006), A snake-like swimming robot using IPMC actuator/sensor, *Proceedings 2006 IEEE International*

- conference on robotics and automation, ICRA*, Editor^Editors, IEEE, Orlando, FL, USA, pp. 1812–1817.
40. Khatib, O. (1996), "Motion coordination and reactive control of autonomous multi-manipulator system", *Journal of Robotic Systems*. 15(4), pp. 300-319.
 41. Khatib, O., Yokoi, K., Brock, O., Chang, K., Casal, A. (1999), "Robots in human environments: Basic autonomous capabilities", *The International Journal of Robotics Research* 18(7), pp. 684-696.
 42. Khatib, Oussama (1986), "Real-time obstacle avoidance for manipulators and mobile robots", *The international journal of robotics research*. 5(1), pp. 90-98.
 43. Kim, D. H., Lee, S. U., Nam, J. H., Furukawa, Y. (2019), "Determination of ship collision avoidance path using deep deterministic policy gradient algorithm", *Journal of the Society of Naval Architects of Korea*. 56(1), pp. 58-65.
 44. Kim, Jae Hyun, SungWook Lee, Eun Seok Jin (2022), "Collision avoidance based on predictive probability using Kalman filter", *International Journal of Naval Architecture and Ocean Engineering*. 14, p. 100438.
 45. KL, Klaassen B and Paap (1999), GMD-SNAKE2: a snake-like robot driven by wheels and a method for motion control, *Proceedings. 1999 IEEE International Conference on Robotics and Automation*, IEEE, Detroit, MI, USA, pp. 3014–3301.
 46. Koren, Yoram, Johann Borenstein (1991), "Potential field methods and their inherent limitations for mobile robot navigation", *Icra*. 2(1991), pp. 1398-1404.
 47. Köseoğlu, Murat, Orkan Murat Çelik, and Ömer Pektaş (2017), Design of an autonomous mobile robot based on ROS, *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, IEEE, pp. 1-5.

48. Krogh, Bruce (1984), A generalized potential field approach to obstacle avoidance control, *Proc. SME Conf. on Robotics Research: The Next Five Years and Beyond*, Bethlehem, pp. 11-22.
49. Krogh, Bruce, Charles Thorpe (1986), Integrated path planning and dynamic steering control for autonomous vehicles, *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, IEEE, pp. 1664-1669.
50. Kuwata, Y., Wolf, M. T., Zarghitzky, D., Huntsberger, T. L. (2013), "Safe maritime autonomous navigation with COLREGS, using velocity obstacles", *IEEE Journal of Oceanic Engineering*. 39(1), pp. 110-119.
51. Kyung Hyun Choi, Chang Jong Kim, Minh Ngoc Nong (2007), Real-time Obstacle Avoidance for Silvermate Robot, *Proceedings of the KSME Conference*, Korea, pp. 1161-1166.
52. Kyung Hyun Choi, Hyoung Chan Yang, Minh Ngoc Nong (2007), A Study on the Real-Time Obstacle Avoidance of Robot based on Elastic Force, *The 13th International Conference on Advanced Robotics ICAR 2007*, Jeju, Korea, pp. 1081-1086.
53. Laporte, François (2014), JAC software, solving conjunction assessment issues, *Advanced Maui Optical and Space Surveillance Technologies Conference*, p. E4.
54. LAVALLE, SM (1998), *Rapidly-exploring random trees: A new tool for path planning*, Computer Science Dept. Oct.
55. Lee, Jang Gyu, Hakyoung Chung (1994), "Global path planning for mobile robot with grid-type world model", *Robotics and computer-integrated manufacturing*. 11(1), pp. 13-21.
56. Lei Chen, Xian-Zong Bai, Kebo Li (2017), "Application of Collision Probability", *Orbital Data Applications for Space Objects*, National Defense Industry Press, Beijing and Springer Nature Singapore Pte Ltd, pp. 185-246.
57. Li, J., Ran, M., Wang, H., & Xie, L. (2019), MPC-based unified trajectory planning and tracking control approach for automated guided vehicles, *2019*

- IEEE 15th International Conference on Control and Automation (ICCA)*, IEEE, pp. 374-380.
58. Liu, Q., Zhao, L., Tan, Z., Chen, W (2017), "Global path planning for autonomous vehicles in off-road environment via an A-star algorithm", *International Journal of Vehicle Autonomous Systems* 13(4), pp. 330-339.
 59. M. Virt, S. Siroki, Á. Nyerges, V. Tihanyi (2019), An Online Path Planning Algorithm for Automated Vehicles for Slow Speed Maneuvering, *2019 International IEEE Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, pp. 97–102.
 60. Mester, Gyula (2006), Motion control of wheeled mobile robots, *4th Serbian-Hungarian Joint Symposium on Intelligent Systems*, Editor^Editors, Hungary, pp. 119-130.
 61. Moravec, Hans, Alberto Elfes (1985), High resolution maps from wide angle sonar, *Proceedings. 1985 IEEE international conference on robotics and automation*, IEEE, pp. 116-121.
 62. Moravec, Hans P. (1988), "Sensor fusion in certainty grids for mobile robots", *AI magazine*. 9(2), pp. 61-61.
 63. Murthy, V. M (2011), "Autonomous mobile robots designing", *Journal of Global Research in Computer Science*. 2(4).
 64. Namgung, H., Jeong, J. S., Kim, J. S., Kim, K. I. (2019), "Inference Model of Collision Risk Index based on Artificial Neural Network using Ship Near-Collision Data", *Journal of Physics: Conference Series*. 1357(1), p. 012044.
 65. Namgung, Ho (2019), "Inference Rule of Collision Risk Index based on Ship Near-Collision via Adaptive Neuro Fuzzy Inference System", *Advances in Science, Technology and Engineering Systems Journal*. 4(4), pp. 152-160.
 66. Newman, W., Neville Hogan (1987), High speed robot control and obstacle avoidance using dynamic potential functions, *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, IEEE, pp. 14-24.

67. Oliver Brock, Oussama Khatib (2002), "Elastic Strips: A Framework for Motion Generation in Human Environments", *The International Journal of Robotics Research*. 21(12).
68. Park, Jinwan, Jung-Sik Jeong (2021), "An estimation of ship collision risk based on relevance vector machine", *Journal of Marine Science and Engineering*. 9(5), p. 538.
69. Peterson, Glenn (2004), NEO orbit uncertainties and their effect on risk assessment, *2004 Planetary Defense Conference: Protecting Earth from Asteroids*.
70. Phala, Kgopotjo Simon Elvis, Anuj Kumar, and Gerhard P. Hancke (2016), "Air quality monitoring system based on ISO/IEC/IEEE 21451 standards", *IEEE Sensors Journal* 16(12), pp. 5037-5045.
71. Rafai, Anis Naema Atiyah, et al. (2022), "A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots", *Journal of Robotics*. 2022, pp. 1-14.
72. Research, Allied Market (2019), *Mobile Robotics Market by Product (UGV, UAV, and AUV), Component (Hardware, Software, and Support & Services), Application (Logistics & Warehousing, Military & Defense, Healthcare, Domestic, Entertainment, Education, Agriculture & Forestry, and Others): Global Opportunity Analysis and Industry Forecast, 2019 - 2026*, UK, accessed May, 15th-2023, from <https://www.alliedmarketresearch.com/mobile-robotics-market>.
73. Research, Precedence (2022), *Robotics Technology Market (By Component: Hardware, Software, Service; By Robot Type: Traditional industrial robots, Cobots, Professional service robots, Others; By Application: Manufacturing, Healthcare, Aerospace & Defense, Media & Entertainment, Logistics, Others) - Global Industry Analysis, Size, Share, Growth, Trends, Regional Outlook, and Forecast 2022-2030*, Canata, accessed August, 2022-2022, from <https://www.precedenceresearch.com/robotics-technology-market>.

74. Rostami, S. M. H., Sangaiah, A. K., Wang, J., Liu, X. (2019), "Obstacle avoidance of mobile robots using modified artificial potential field algorithm", *EURASIP Journal on Wireless Communications and Networking*. 2019(1), pp. 1-19.
75. Rubio, Francisco, Francisco Valero, Carlos Llopi-Albert (2019), "A review of mobile robots: Concepts, methods, theoretical framework, and applications", *International Journal of Advanced Robotic Systems*. 16(2), p. 1729881419839596.
76. Soto, Fernando, et al (2020), "Medical micro/nanorobots in precision medicine.", *Advanced Science*. 7(21), p. 2002203.
77. Wang, Yaoyao, et al. (2017), "A new redundancy resolution for underwater vehicle–manipulator system considering payload", *International Journal of Advanced Robotic Systems* 14(5).
78. Woo-Joohyun, Nakwan-Kim (2020), "Collision avoidance for an unmanned surface vehicle using deep reinforcement learning", *Ocean Engineering* 199, p. 107001.
79. Xu, H., Yu, Z., Lu, X., Wang, S., Li, S., Wang, S. (2020), Model predictive control-based path tracking control for automatic guided vehicles, *2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, IEEE, pp. 627-632.
80. Zhou, C., Wang, Y., Wang, L., He, H (2022), "Obstacle avoidance strategy for an autonomous surface vessel based on modified deep deterministic policy gradient", *Ocean Engineering*. 24(3), p. 110166.
81. Bùi, Quang Sum (2016), *Xử lý va chạm trong thực tại ảo và ứng dụng*, Học viện Công nghệ Bru chính viễn thông.
82. Đỗ, Thị Chi (2014), *Nghiên cứu một số kỹ thuật tính toán va chạm trong thực tại ảo*, Đại học Quốc gia Hà Nội.

83. Duong, Ngoc Lan Ngo, Van Thuyen (2010), "Potential field method: obstacle avoidance for mobile robots", *Journal of Technical Education Science*(13), pp. 9-16.
84. Nguyễn Đức Hoàng, Đỗ Năng Toàn, Nông Minh Ngọc (2015), Một kỹ thuật xây dựng hệ bao tự động cho đối tượng 3D, *Kỷ yếu hội nghị khoa học công nghệ quốc gia lần thứ VIII. FAIR- Nghiên cứu cơ bản và ứng dụng công nghệ thông tin*, Việt Nam, p. 11.
85. Hưng, Phạm Duy Triển, Phạm Minh (2012), Applied Artificial Potential Field Method for Collision Avoidance Problem in Multi-Robot System, *Hội thảo toàn quốc về Điện tử - Truyền thông – An toàn thông tin, ATC/REV-2012*, Viet Nam.
86. Nông Minh Ngọc, Phạm Đình Lâm, Đỗ Năng Toàn (2014), "Phương pháp phát hiện va chạm dựa trên cấu trúc cây phân cấp giới hạn biên ứng dụng kỹ thuật hộp bao theo hướng", *Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên*. 14, p. 11.
87. Đỗ Năng Toàn, Nông Minh Ngọc (2014), "Tính toán va chạm sử dụng kỹ thuật hộp bao theo hướng và Ứng dụng trong tuyên truyền giao thông", *Tạp chí Khoa học và Công nghệ Đại học Thái Nguyên*. 6, p. 10.
88. Van, Hung Nghiem. Nguyen, Van Can. Hoang, Viet Long. Trinh, Hien Anh (2022), "Kỹ thuật phát hiện nhanh va chạm của chất liệu vải tương tác trong môi trường thực tại ảo", *Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông*, pp. 49-56.