

Nghiên cứu phương pháp tối ưu hóa dẫn đường toàn cục cho robot tự hành dạng phi đối xứng

Nguyễn Ngọc Tú*

Trung tâm Quang điện tử, Viện Ứng dụng công nghệ, C6 Thanh Xuân Bắc, Thanh Xuân, Hà Nội, Việt Nam.

*Tác giả liên hệ: ngoctu@cfoc.vn

Nhận bài: 23/9/2024; Hoàn thiện: 25/11/2024; Chấp nhận đăng: 28/11/2024; Xuất bản: 25/12/2024.

DOI: <https://doi.org/10.54939/1859-1043.j.mst.100.2024.46-53>

TÓM TẮT

Nghiên cứu này tập trung vào việc tối ưu hóa dẫn đường toàn cục cho robot tự hành (AMR) có cấu trúc và khả năng cơ động không đối xứng, thường gặp trong các ứng dụng thực tế. Nhằm đề xuất một phương pháp tối ưu hóa quỹ đạo toàn cục, giúp AMR dạng phi đối xứng di chuyển hiệu quả trong môi trường phức tạp, giảm thời gian thực hiện nhiệm vụ và tối ưu hóa hiệu suất hoạt động. Nghiên cứu sử dụng thuật toán SPBL để tìm kiếm đường đi tối ưu trong môi trường dựa trên cơ sở quỹ đạo khởi thủy. Quá trình dẫn đường sử dụng DWB được tối ưu hóa bằng cách điều chỉnh các thông số chi phí cho các thao tác cơ bản như di chuyển thẳng, quay tại chỗ, trượt, lùi. Các thông số này được thiết kế để ưu tiên các quỹ đạo trơn, tránh các thao tác phức tạp và không hiệu quả. Kết quả của nghiên cứu cho thấy phương pháp đề xuất giúp giảm thiểu độ phức tạp của quỹ đạo, dẫn đến việc giảm thời gian thực hiện nhiệm vụ và cải thiện hiệu suất hoạt động của AMR dạng phi đối xứng, có khả năng ứng dụng trong các hệ thống vận chuyển bằng robot trong công nghiệp.

Từ khóa: AMRs; Robot tự hành; Dẫn đường toàn cục; DWB; SPBL.

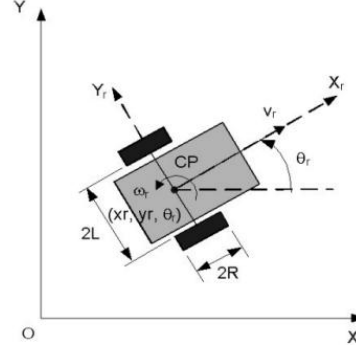
1. MỞ ĐẦU

Nghiên cứu phương pháp tối ưu hóa dẫn đường toàn cục cho Robot tự hành dạng phi đối xứng (AMR) là một lĩnh vực nghiên cứu trong ngành robot nhằm phát triển các phương pháp giúp Robot di chuyển từ điểm xuất phát đến điểm đích một cách hiệu quả và an toàn. Một trong những nhiệm vụ quan trọng là tối ưu hóa dẫn đường toàn cục để giải quyết vấn đề lập kế hoạch tuyến đường cụ thể cho Robot trên một khoảng cách lớn, đảm bảo rằng Robot có thể di chuyển thông qua một loạt các địa điểm một cách thông minh, nhanh chóng và tránh xa các chướng ngại vật hoặc vùng cấm. Với AMR dạng phi đối xứng Robot không thể di chuyển đối xứng với trục của nó, nó cần phải có các phương pháp tối ưu hóa dẫn đường đặc biệt để giải quyết vấn đề di chuyển. Trong nghiên cứu do T. R. J. Hrbáček và các cộng sự (2010) [1] đưa ra phương pháp tối ưu hóa dẫn đường cho robot Ackermann được thực hiện thông qua việc kiểm soát tốc độ góc của các bánh sau bằng một bộ vi sai phần mềm. Theo nghiên cứu [1], tốc độ góc của các động cơ được điều chỉnh dựa trên góc lái hiện tại và tốc độ tiến tới mong muốn, giúp robot hoạt động hiệu quả về năng lượng và đảm bảo tính linh hoạt khi di chuyển trên các địa hình khó khăn. Tuy nhiên, robot có thể trở nên khó kiểm soát hơn nếu một trong các đơn vị dẫn động bị lỗi. Tương tự, trong nghiên cứu của Wang (2018)[2] trình bày phương pháp điều khiển dự đoán mô hình (MPC) để tối ưu hóa dẫn đường cho robot omni. Phương pháp này sử dụng mô hình động học của robot để dự đoán hành vi tương lai và tối ưu hóa quỹ đạo dựa trên các ràng buộc và mục tiêu đã định sẵn. MPC [3] cho phép điều chỉnh liên tục quỹ đạo di chuyển của robot để tránh chướng ngại vật và đạt đến điểm đích một cách tối ưu. Phương pháp này tích hợp thuật toán tìm đường A* để xác định quỹ đạo ban đầu, sau đó tinh chỉnh bằng MPC để đảm bảo tính khả thi và tối ưu trong thời gian thực. Kết quả thử nghiệm [3] cho thấy hệ thống MPC có khả năng điều khiển robot omni di chuyển mượt mà và chính xác trong môi trường phức tạp những kết quả nghiên cứu trước về vấn đề cần nghiên cứu [1-3].

2. PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Mô hình động học của Robot

Mô hình [4] của một robot di động có bánh xe được trình bày trong hình 1. Mô hình này tính đến hai bánh xe dẫn động nằm đối diện nhau có bán kính R , khoảng cách giữa hai bánh xe ($2L$), tốc độ góc của các bánh xe dẫn động (ω_L, ω_R), và điểm trung tâm (CP) của robot. Tư thế [4] của robot được coi là $P = (x_r, y_r, \theta_r)$, trong đó, x_r đại diện cho vị trí trên trục x , y_r đại diện cho vị trí trên trục y và θ_r đại diện cho hướng của robot. v_r đại diện cho vận tốc tuyến tính trong khi ω_r đại diện cho vận tốc góc của robot.



Hình 1. Sơ đồ động học của robot [4]

Mô hình động học cho robot có thể được biểu diễn hệ phi tuyến như sau:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} \quad (1)$$

[4] Mô hình động học của robot cũng có thể được biểu diễn:

$$\begin{cases} \dot{x}_r = v_r \cos \theta_r \\ \dot{y}_r = v_r \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases} \quad (2)$$

2.2. Thuật toán EKF

Nhiệm vụ của robot AMR là tự động điều hướng trong các môi trường có cấu trúc. Để làm được việc đó, AMR cần ước lượng chính xác vị trí và hướng, bao gồm sự kết hợp dữ liệu thu thập từ các cảm biến như encoder gắn trên robot, IMU. Do trượt và nhiễu, ước lượng vị trí và hướng chỉ dựa vào hai encoder không đáng tin cậy. Hơn nữa, việc ước lượng vị trí và hướng sử dụng IMU chỉ đáng tin cậy trong thời gian ngắn do các lỗi hệ thống. Trong trường hợp này, EKF được sử dụng để giảm nhiễu và sai lệch của cảm biến [5]. Kalman Filter (KF): Là thuật toán tối ưu để ước lượng trạng thái của một hệ thống tuyến tính khi biết các phép đo có chứa nhiễu. KF sử dụng mô hình trạng thái tuyến tính và nhiễu Gaussian. Extended Kalman Filter (EKF): Khi hệ thống là phi tuyến, KF không còn hoạt động hiệu quả nữa. EKF giải quyết vấn đề này bằng cách tuyến tính hóa mô hình phi tuyến xung quanh ước lượng hiện tại (Extended Kalman filter, n.d.) [6].

Trong bộ lọc Kalman mở rộng, mô hình chuyển trạng thái và mô hình quan sát không cần phải là các hàm tuyến tính của trạng thái mà có thể là các hàm khả vi.

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}) + w_{k-1} \\ z_k &= h(x_k) + v_k \end{aligned} \quad (3)$$

Trong đó, w_k và v_k là nhiễu quá trình và nhiễu đo lường, cả hai đều được giả định là nhiễu Gaussian đa biến có trung bình bằng không với ma trận hiệp phương sai lần lượt là Q_k và R_k . u_k là vector điều khiển. Hàm f có thể được sử dụng để tính toán trạng thái dự đoán từ ước lượng trước đó và tương tự hàm h có thể được sử dụng để tính toán giá trị đo lường dự đoán từ trạng thái dự đoán. Tuy nhiên, f và h không thể áp dụng trực tiếp lên ma trận hiệp phương sai. Thay vào đó, một ma trận đạo hàm riêng phần (Jacobian) được tính toán. Tại mỗi bước thời gian, ma trận Jacobian được tính toán với các trạng thái dự đoán hiện tại. Các ma trận này có thể được sử dụng trong các phương trình của bộ lọc Kalman. Quá trình này thực chất là tuyến tính hóa hàm phi tuyến

xung quanh ước lượng hiện tại.

2.3 Thuật toán xây dựng bản đồ dựa trên SLAM TOOLBOX

Các thuật toán SLAM có thể được phân loại thành hai nhóm chính: các thuật toán trước đây sử dụng phương pháp lọc Bayes (Thrun et al., 2005) [7], và các phương pháp dựa trên đồ thị mới hơn (Thrun & Montemerlo, 2006) [8]. Các triển khai nổi bật dựa trên bộ lọc có sẵn dưới dạng gói ROS bao gồm GMapping (Grisetti et al., 2007) [9] và HectorSLAM (Kohlbrecher et al., 2011) [10]. Các triển khai dựa trên đồ thị chính có sẵn là Cartographer (Hess et al., 2016) [11] và KartoSLAM (Konolige et al., 2010) [12] (Steve Macenski, 2021) [13].

2.3.1. Thuật toán SLAM Dựa Trên Bộ Lọc

Gmapping [9], sử dụng phương pháp lọc hạt để xây dựng bản đồ lưới từ dữ liệu LiDAR 2D, thuật toán này không phù hợp cho các không gian lớn và do không thể đóng vòng lặp chính xác. HectorSLAM [10] dựa vào việc khớp quét LiDAR và bộ lọc định hướng 3D dựa trên ước lượng trạng thái EKF. Phương pháp này tập trung vào việc ước lượng vị trí robot trong thời gian thực và tạo ra bản đồ 2D với tốc độ cập nhật cao. HectorSLAM [10] không sử dụng dữ liệu quán tính, điều này có thể dẫn đến việc ước lượng vị trí và cập nhật bản đồ không chính xác khi LiDAR ở tốc độ thấp hoặc khi lập bản đồ các không gian lớn hoặc ít đặc điểm. HectorSLAM [10] cũng không cung cấp khả năng đóng vòng lặp, làm cho nó không phù hợp cho việc lập bản đồ đáng tin cậy của các không gian lớn hoặc khi sử dụng máy quét Laser với tốc độ cập nhật thấp (Steve Macenski, 2021) [13].

2.3.2. Thuật toán SLAM Dựa Trên đồ thị

KartoSLAM [12] và Cartographer [11] đều là các thuật toán dựa trên đồ thị, lưu trữ một đồ thị các vị trí và đặc điểm của Robot. Các thuật toán dựa trên đồ thị chỉ cần duy trì đồ thị vị trí, điều này thường làm cho chúng hiệu quả trong việc quản lý tài nguyên, đặc biệt là khi xây dựng bản đồ ở quy mô lớn. Slam Toolbox với tính năng SLAM hoàn chỉnh được xây dựng trên bộ ghép mẫu quét tăng cường tại trung tâm của Karto đã được nâng cấp và tăng tốc để sử dụng trong gói này [13]. Phương pháp định vị mới có tên "định vị đồ thị đàn hồi" được đưa ra, lấy một cửa sổ trượt của các phép đo và thêm chúng vào đồ thị để tối ưu hóa và tinh chỉnh. Điều này cho phép theo dõi các đặc điểm địa phương đã thay đổi trong môi trường thay vì coi chúng là những sai lệch và loại bỏ các nút thừa khi robot rời khỏi khu vực để không làm ảnh hưởng đến bản đồ dài hạn.

2.4. Thuật toán SPBL lattice planner

Thuật toán SBPL lattice planner [15, 16] là một thuật toán lập kế hoạch di chuyển cho robot di chuyển trong một môi trường đa chiều bằng cách sử dụng lưới (lattice) để biểu diễn môi trường và lên kế hoạch. Lattice planner được xây dựng bằng cách sử dụng một bộ sinh đường đi nghịch đảo để tìm các đường đi giữa bất kỳ nút nào trong lưới và nguồn gốc, có hai tính chất quan trọng (Milano, 2018) [17]: *Rời rạc hóa*: chuyển đổi vấn đề lập kế hoạch di chuyển thành một quá trình quyết định tuần tự. Giả sử rằng các quyết định chỉ được thực hiện tại các trạng thái rời rạc, thì các trạng thái này là các nút trong lưới và các chuyển động là các cạnh nối các trạng thái đó. *Tính Quy tắc*: Nếu quá trình rời rạc hóa có bất kỳ mức độ quy tắc nào thì lưới trạng thái sẽ có tính chất bất biến dịch chuyển, nghĩa là có thể lặp lại cùng một chuyển động ở nhiều nút của lưới. Xét rằng lưới là quy tắc trong các tọa độ dịch chuyển (x,y), thì cùng một chuyển động từ trạng thái A đến B cũng có thể dẫn từ trạng thái C đến D. Chính xác hơn, nếu đường đi giữa hai tư thế là khả thi:

$$\begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \end{pmatrix} \rightarrow \begin{pmatrix} x_2 \\ y_2 \\ \theta_2 \end{pmatrix} \quad (4)$$

Thì đường đi sau đây cũng khả thi với bất kỳ giá trị nào của số nguyên k:

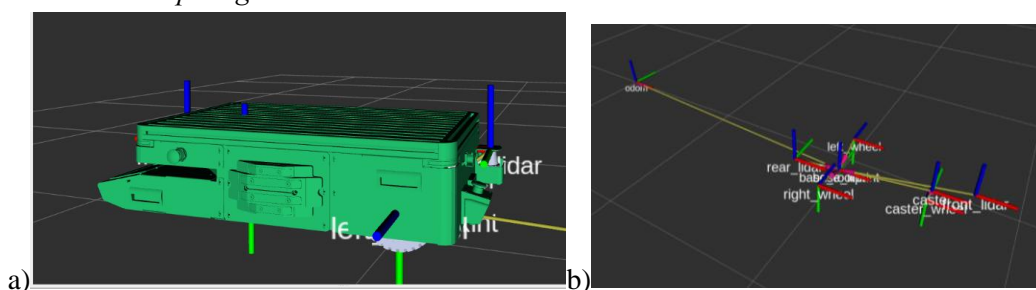
$$\begin{pmatrix} x_1 + k\Delta_l \\ y_1 + k\Delta_l \\ \theta_1 \end{pmatrix} \rightarrow \begin{pmatrix} x_2 + k\Delta_l \\ y_2 + k\Delta_l \\ \theta_2 \end{pmatrix} \quad (5)$$

2.5 Thuật toán DWB (Dynamic Windows Approacher)

DWB (Dynamic Window Approaches) Local Planner [19] là một thuật toán quyết định di chuyển robot trong môi trường đã biết. Thuật toán này sử dụng cửa sổ động để tạo ra các tập hợp các tốc độ và hướng di chuyển tiềm năng cho robot, sau đó lựa chọn tốc độ và hướng di chuyển phù hợp dựa trên các ràng buộc và mục tiêu di chuyển. Thuật toán DWB Local Planner xác định các điểm cuối tiềm năng (trong một bản đồ đã biết) cho robot và tập hợp các hướng di chuyển có thể đạt được từ vị trí hiện tại của robot đến các điểm cuối đó. Từ đó, nó tạo ra một cửa sổ động (dynamic window) bằng cách kết hợp các thông số như tốc độ tối đa và tốc độ xoay tối đa của robot để giới hạn các tùy chọn di chuyển. Sau khi có được cửa sổ động, thuật toán sẽ đánh giá các tốc độ và hướng di chuyển có thể xem xét trong cửa sổ đó, và chọn ra tốc độ và hướng di chuyển tối ưu dựa trên mục tiêu di chuyển của robot và các ràng buộc như tránh vật cản.

2.6. Chuẩn bị thực nghiệm

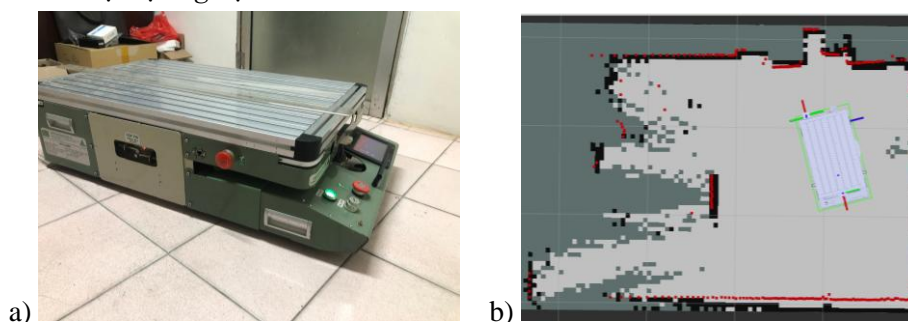
2.6.1. Mô hình mô phỏng robot



Hình 2. Mô hình mô phỏng (a) và TF (transform) của đế robot phi đối xứng (b).

Mô hình TF trong ROS được sử dụng để theo dõi các khung tọa độ (coordinate frames) và các phép biến đổi (transformations) giữa chúng. Điều này rất quan trọng trong robot để quản lý và chuyển đổi giữa các không gian tọa độ khác nhau, chẳng hạn như từ khung tọa độ của robot sang khung tọa độ của một cảm biến gắn trên robot – hình 2. Mỗi khung tọa độ (TF) đều có vai trò quan trọng trong việc định vị và điều khiển robot, giúp robot tương tác chính xác với môi trường.

2.6.2. Thiết kế thực nghiệm



Hình 3. Robot thực nghiệm (a) và biểu diễn trên RVIZ cùng với bản đồ đã xây dựng (b).

Tiến hành thực nghiệm trên Robot phi đối xứng trong môi trường thực tế theo hình 3 và cây TF thiết lập cho các bộ phận của Robot tới gốc tọa độ của bản đồ; cài đặt các thông số cho các mô đun xử lý tính toán.

3. MÔ PHỎNG, TÍNH TOÁN, THẢO LUẬN

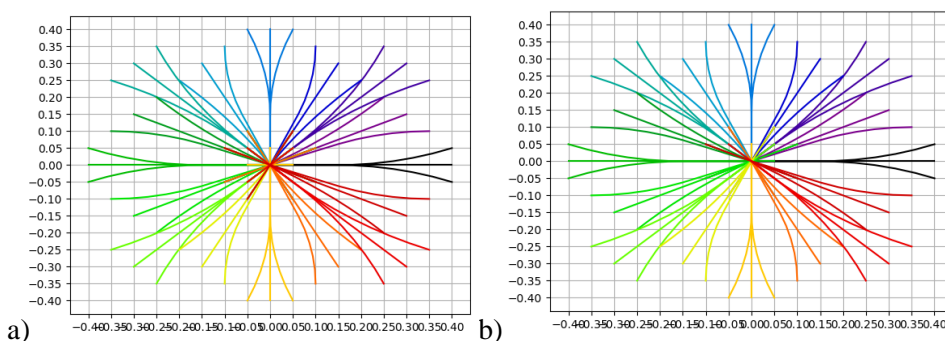
3.1. Thiết lập thông số tạo quỹ đạo nguyên thủy (mprim) cho SPBL

Kích thước và độ phân giải của lattice: Kích thước và độ phân giải của lattice ảnh hưởng trực tiếp đến độ chính xác và độ phức tạp của quá trình lập kế hoạch. Tập hợp các hành động có thể

thực hiện được (mprim): Xác định các hành động khả thi dựa trên khả năng cơ động của robot; Chi phí của các hành động (mprim): Gán chi phí cho mỗi hành động để ưu tiên các đường đi ngắn nhất, nhanh nhất hoặc hiệu quả nhất. Có 2 bộ thông số thiết lập cho robot đối xứng và phi đối xứng: Bảng 1 là chi phí cho các hướng di chuyển hình 4, bảng 2 là thông số thiết lập cho thuật toán thực thi SBPLLatticePlanner, để thực nghiệm đánh giá tính phù hợp của hoạch định chuyển động toàn cục đối với cùng một Robot thực thể và phản ứng khi robot di chuyển đạt được tối ưu.

Bảng 1. Bảng thông số thiết lập cho mprim.

STT	Tên thông số	Robot đối xứng	Robot phi đối xứng
1	Resolution (m)	0,05	0,05
2	Numberofangles	16	8
3	nNberofprimspersangle	5	9
4	Forwardcostmult	1	1
5	Backwardcostmult	5	40
6	Forwardandturncostmult	2	1
7	Sidestepcostmult	50	10
8	Turninplacecostmult	5	20



Hình 4. Biểu đồ xu hướng di chuyển và chi phí cho mprim a) robot đối xứng, b) phi đối xứng.

Bảng 2. Bảng thông số thiết lập cho SBPLLatticePlanner.

STT	Thông số	Giá trị
1	environment_type	XYThetaLattice
2	planner_type	ARAPlanner
3	allocated (s)	50
4	initial_epsilon	3,0
5	nominalvel_mpersecs (m/s)	0,4
6	forward_search	False
7	timetoturn45degsinplace_secs	1,97
8	lethal_obstacle	20

3.2. Thiết lập thông số cho lập kế hoạch cục bộ DWB (local plainning)

Để robot có thể thực hiện bám theo quỹ đạo toàn cục, cần thực hiện các bước cấu hình phù hợp với thông số và đặc tính của robot. Vai trò của bộ đánh giá trong DWB: Đánh giá các đường đi khả thi: Bộ đánh giá nhận đầu vào là một tập hợp các đường đi khả thi (tạo ra bởi bộ tạo đường đi) và đánh giá từng đường đi dựa trên một số tiêu chí nhất định; Xếp hạng các đường đi: Bộ đánh giá sẽ xếp hạng các đường đi theo độ ưu tiên, dựa trên kết quả đánh giá; Chọn đường đi tốt nhất: Đường đi có thứ hạng cao nhất (thường là đường đi tốt nhất theo các tiêu chí đánh giá) sẽ được chọn để robot thực hiện. Bảng 3 là các thiết lập cho move_base thực thi quá trình điều khiển robot,

bảng 4 thiết lập các giới hạn về tốc độ tịnh tiến và quay tròn của DWB điều khiển, hoạch định chuyển động cục bộ, bảng 5 là thông số các bộ đánh giá tối ưu quỹ đạo và phân chia các tư thế khi bắt đầu quỹ đạo.

Bảng 3. Bảng thông số thiết lập cho move_base.

STT	Thông số	Giá trị
1	Footprint (m)	[[0,89,-0,31],[0,89,0,31],[-0,21,0,31],[-0,21,-0,31]]
2	controller_frequency (Hz)	10
3	controller_patience (s)	15
4	planner_frequency (Hz)	0
5	planner_patience (s)	10,0
6	max_planning_retries	5
7	oscillation_timeout	15
8	oscillation_distance (m)	0,1

Bảng 4. Bảng thông số thiết lập cho DWB.

STT	Thông số	Giá trị	STT	Thông số	Giá trị
1	max_vel_x (m/s)	0,9	9	acc_lim_x (m ² /s)	0,08
2	min_vel_x (m/s)	-0,6	10	acc_lim_y (m ² /s)	0
3	max_vel_y (m/s)	0	11	acc_lim_theta(rad ² /s)	0,25
4	min_vel_y (m/s)	0	12	decel_lim_x(m ² /s)	-0,08
5	max_speed_xy (m/s)	0,9	13	decel_lim_y	0
6	min_speed_xy (m/s)	0,05	14	decel_lim_theta(rad ² /s)	-0,25
7	max_vel_theta (rad/s)	1,0	15	yaw_goal_tolerance (rad)	0,08
8	min_speed_theta (rad/s)	0,05	16	xy_goal_tolerance(m)	0,1

Bảng 5. Bảng thông số thiết lập các bộ đánh giá sử dụng cho DWB.

STT	Bộ đánh giá (Critics)	Thông số	Giá trị
1	GoalAlign	scale	24
		forward_point_distance (m)	0,1
2	RotateToGoal	scale	32
		lookahead_time	-1,0
		slowing_factor	5,0
3	PathDistPruned	scale	32
4	PathProgress	scale	24,0
		heading_scale	0,1
		xy_local_goal_tolerance(m)	0,01
		angle_threshold (rad)	0,3488

3.3. Phương pháp, công cụ mô phỏng

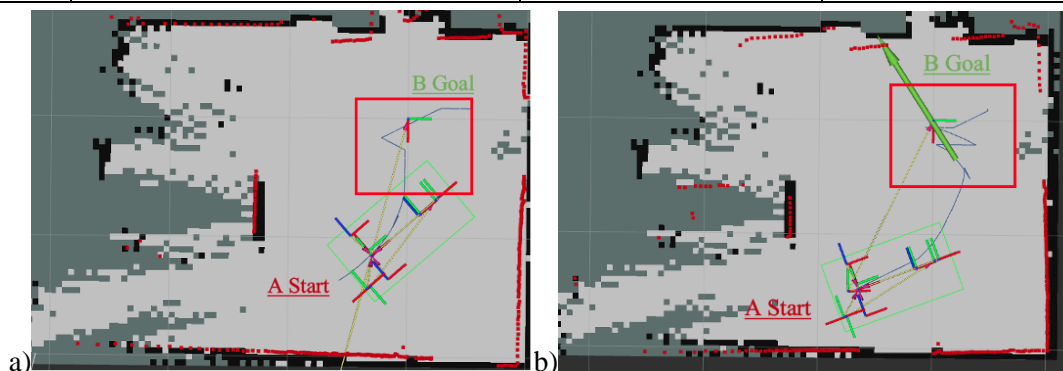
Thiết lập điểm đích (goal) cho robot từ vị trí A tới vị trí B theo hai kiểu quỹ đạo nguyên thủy mprim trong SPBL khác nhau để robot tính toán quỹ đạo đường đi toàn cục, sau đó xác định độ phức tạp của quỹ đạo sinh ra và thời gian thực hiện.

3.4. Kết quả và bình luận

Hình 5 -a là thể hiện quỹ đạo khi thực hiện với mprim cho phi đối xứng, và hình 5-b cho mprim đối xứng từ điểm xuất phát **A Start tới B Goal**, và bảng 6 là số đoạn thể hiện tính phức tạp của quỹ đạo toàn cục và thời gian robot thực hiện.

Bảng 6. Bảng thông số thực nghiệm thực hiện nhiệm vụ di chuyển của robot.

STT	Thông số	Quỹ đạo phi đối xứng	Quỹ đạo đối xứng
1	Số đoạn được hình thành	4	7
2	Thời gian thực hiện nhiệm vụ (giây)	15	32



Hình 5. Quỹ đạo đường đi của tối ưu cho dạng phi đối xứng a) và dạng đối xứng b).

Kết quả thực nghiệm cho thấy việc tối ưu hóa quỹ đạo nguyên thủy có ảnh hưởng đáng kể đến việc tạo ra quỹ đạo toàn cục dẫn đường cho robot. Hình 5-a và 5-b minh họa các đường gấp khúc của quỹ đạo trong pha cuối khi robot tiếp cận điểm mục tiêu B trong không gian hẹp. Bằng cách tăng chi phí cho các thao tác lùi, trượt và quay tại chỗ, đồng thời giảm số góc chuyển hướng, độ phức tạp của quỹ đạo được giảm thiểu. Điều này giúp robot hoàn thành tác vụ di chuyển nhanh chóng hơn, rút ngắn thời gian thực hiện nhiệm vụ.

4. KẾT LUẬN

Nghiên cứu này đã đề xuất phương pháp để thiết lập tối ưu cho lập kế hoạch toàn cục (SPBL), dẫn đường (DWB) và định vị (Slam-toolbox) đối với robot tự hành (AMR) dạng phi đối xứng. Trong đó, các thông số quan trọng trong quỹ đạo khởi thủy mprim SPBL có tác động tới độ phức tạp của quỹ đạo toàn cục, việc quy định số quỹ đạo và hướng di chuyển là yếu tố then chốt; Bên cạnh đó, việc thiết lập các bộ đánh giá (Critics) trong DWB giúp robot dễ dàng bắt bám quỹ đạo toàn cục và phân chia điều khiển theo các đoạn và điểm đích tạm thời theo các khu vực chọn lọc tiến- quay, chỉ quay, lùi,... Trong hướng nghiên cứu sắp tới, nhóm nghiên cứu sẽ tập trung vào giải quyết vấn đề làm trơn mịn đường quỹ đạo toàn cục, giảm bớt các đường cắt, gấp khúc và hạn chế các điểm đích tạm thời nhằm giúp robot di chuyển dễ dàng hơn.

Lời cảm ơn: Tác giả cảm ơn sự tài trợ về kinh phí từ đề tài “Nghiên cứu, xây dựng hệ thống quản lý và điều phối các robot tự hành vận chuyển trong công nghiệp”.

TÀI LIỆU THAM KHẢO

- [1]. Hrbáček, J., T. Ripel, and J. Krejsa. "Ackermann mobile robot chassis with independent rear wheel drives." Proceedings of 14th International Power Electronics and Motion Control Conference EPE-PEMC 2010. IEEE, (2010).
- [2]. Wang, C., Liu, X., Yang, X., Hu, F., Jiang, A., & Yang, C. "Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy". Applied Sciences, 8(2), 231, (2018).
- [3]. Tătar, Mihai Olimpiu, et al. "Design and development of an autonomous omni-directional mobile robot with Mecanum wheels." 2014 IEEE International Conference on Automation, Quality and Testing, Robotics. IEEE, (2014).
- [4]. V. M. B. D. A. F. A. Filipescu, "Trajectory-tracking and discrete-time sliding-mode control of wheeled mobile robots," 2011. IEEE International Conference on Information and Automation. IEEE, 2011.
- [5]. Z. J. Shibing Yu, "Design of the navigation system through the fusion of IMU and wheeled encoders," 2020. Computer Communications 160: 730-737, (2020).

- [6]. Kalman, Rudolph E., and Richard S. Bucy. "New results in linear filtering and prediction theory.": 95-108, (1961).
- [7]. Thrun, S.. "Simultaneous localization and mapping. In *Robotics and cognitive approaches to spatial mapping*", (pp. 13-41). Berlin, Heidelberg: Springer Berlin Heidelberg, (2008).
- [8]. Thrun, S., & Montemerlo, M. "The graph SLAM algorithm with applications to large-scale mapping of urban structures". *The International Journal of Robotics Research*, 25(5-6), 403-429, (2006).
- [9]. Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters", *IEEE Transactions on Robotics*, Volume 23, pages 34-46, (2007).
- [10]. Kohlbrecher, S., Meyer, J., Graber, T., Petersen, K., Klingauf, U., & Von Stryk, O.. "Hector open source modules for autonomous mapping and navigation with rescue robots". In *RoboCup 2013: Robot World Cup XVII 17*, pp. 624-631. Springer Berlin Heidelberg, (2013).
- [11]. Hess, W., Kohler, D., Rapp, H., & Andor, D. "Real-time loop closure in 2D LIDAR SLAM". In 2016 IEEE international conference on robotics and automation (ICRA), pp. 1271-1278, (2016).
- [12]. Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., & Vincent, R. "Efficient sparse pose adjustment for 2D mapping". In 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 22-29, (2010).
- [13]. I. J. Steve Macenski, "SLAM Toolbox: SLAM for the dynamic world", (2021).
- [14]. Agarwal, S., Mierle, K., & Others. (n.d.). Ceres solver. <http://ceres-solver.org>
- [15]. Likhachev, Maxim. "Search-Based Planning Library (SBPL)." (2016).
- [16]. Limpert, Nicolas, Stefan Schiffer, and Alexander Ferrein. "A local planner for Ackermann-driven vehicles in ROS SBPL." 2015. Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech). IEEE, (2015).
- [17]. A. & R. L. L. d. I. A. e. R. d. P. d. Milano, "Implementation, comparison, and advances in global planners using Ackerman motion primitives," (2018).
- [18]. ROS, "sbpl_lattice_planner," [Online]. Available: https://wiki.ros.org/sbpl_lattice_planner.
- [19]. Fox, D., Burgard, W., & Thrun, S. "The dynamic window approach to collision avoidance". *IEEE Robotics & Automation Magazine*, 4(1), 23-33, (1997).

ABSTRACT

Optimization of global path planning for asymmetric mobile robots

This research focuses on optimizing global path planning for asymmetric mobile robots (AMRs), which are characterized by an asymmetric structure and mobility capabilities commonly encountered in practical applications. The objective is to propose a method for optimizing global trajectory planning, enabling efficient movement of asymmetric AMRs in complex environments, reducing task execution time, and optimizing operational performance. The study employs the SPBL algorithm to search for an optimal path in an environment based on an initial trajectory. The path following process utilizing the Dynamic Window Approach (DWA) is optimized by adjusting cost parameters for basic maneuvers such as straight-line motion, rotations, sliding, and reversing. These parameters are designed to prioritize smooth trajectories while avoiding complex and inefficient maneuvers. The results of the research demonstrate that the proposed method minimizes trajectory complexity, leading to reduced task execution time and improved operational performance of asymmetric AMRs. The findings have potential applications in robotic transport systems in the industrial sector.

Keywords: AMRs; Asymmetric mobile robots; Global path planning; DWB; SPBL.