



DOI:10.22144/ctujos.2026.075

ĐIỀU KHIỂN DỰ BÁO PHI TUYẾN CHO BÁM QUỲ ĐẠO VÀ TRÁNH VẬT CẢN ĐỘNG TRÊN ROBOT DI ĐỘNG LÁI VI SAI

Nguyễn Ngọc Thiện, Đoàn Lê Bảo Duy, Huỳnh Nguyễn Thanh Duy và Trần Đức Thiện*

Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh, Việt Nam

*Tác giả liên hệ (Corresponding author): thientd@hcmute.edu.vn

Thông tin chung (Article Information)

Nhận bài (Received): 10/12/2025

Sửa bài (Revised): 12/01/2026

Duyệt đăng (Accepted): 14/04/2026

Title: Nonlinear model predictive control for trajectory tracking and dynamic obstacle avoidance in differential-drive mobile robots

Author(s): Nguyen Ngoc Thien, Doan Le Bao Duy, Huynh Nguyen Thanh Duy and Tran Duc Thien*

Affiliation(s): Ho Chi Minh City University of Technology and Education, Viet Nam

TÓM TẮT

Trong bài báo này, một phương pháp điều khiển dự báo mô hình phi tuyến (NMPC) cho robot di động hai bánh vi sai đã được đề xuất nhằm bảo đảm bám quỹ đạo và tránh vật cản động. Khung dự báo phi tuyến giúp xử lý tính phi tuyến của mô hình robot, nâng cao độ chính xác quỹ đạo và độ ổn định điều hướng. Bằng cách tích hợp các ràng buộc vận tốc tuyến tính và vận tốc góc, phương pháp mang lại chuyển động mượt mà và đáng tin cậy hơn. Các ràng buộc khoảng cách an toàn được đưa vào bài toán tối ưu NMPC để đảm bảo di chuyển an toàn, không va chạm trong môi trường động. Các mô phỏng trên mô hình robot 3D trong Gazebo (ROS) và thí nghiệm trên robot hai bánh thực tế được tiến hành và so sánh với phương pháp DWA. Kết quả cho thấy NMPC vượt trội hơn về độ chính xác bám quỹ đạo, độ mượt của chuyển động và hiệu quả tránh vật cản.

Từ khóa: Bám quỹ đạo, điều khiển dự báo mô hình (MPC), điều khiển dự báo phi tuyến (NMPC), môi trường động, robot di động lái vi sai, tránh va chạm

ABSTRACT

This paper presents a nonlinear model predictive control (NMPC) scheme for differential-drive mobile robots to achieve trajectory tracking and dynamic obstacle avoidance. The nonlinear predictive framework effectively addresses robot model nonlinearities, enhancing trajectory accuracy and navigation stability. By integrating input constraints on linear velocity and angular velocity, the method ensures smoother and more reliable motion. Safety distance constraints are embedded into the NMPC optimization to guarantee collision-free navigation in dynamic environments. Simulations on a 3D robot model in Gazebo under ROS, combined with experiments on a real two-wheeled robot, are conducted and compared with the Dynamic Window Approach (DWA). Results confirm that NMPC outperforms DWA in trajectory tracking accuracy, motion smoothness, and obstacle avoidance effectiveness.

Keywords: Collision avoidance, differential-drive mobile robot, dynamic environments, model predictive control (MPC), nonlinear MPC, trajectory tracking

1. GIỚI THIỆU

Trong những năm gần đây, robot di động hai bánh lái vi sai (di chuyển bằng cách quay hai bánh với vận tốc khác nhau) đã được ứng dụng rộng rãi trong các nhiệm vụ tự động, bao gồm vận chuyển, khảo sát và cứu hộ. Tuy nhiên, việc duy trì bám quỹ đạo ổn định và tránh vật cản động trong môi trường phức tạp vẫn là một thách thức lớn đối với các hệ thống điều khiển robot này. Các phương pháp điều khiển truyền thống như DWA (Dynamic Window Approach), APFC (Advanced Potential Field Controller), Q-learning (Quality learning), PID (Proportional Integral Derivative) đã chứng minh được hiệu quả trong việc tránh va chạm, nhưng chúng vẫn gặp khó khăn trong việc duy trì độ chính xác cao khi bám quỹ đạo, đảm bảo thời gian thực trong quá trình hoạt động, đặc biệt là trong môi trường động (Konar et al., 2013; Li et al., 2017; Malone et al., 2017; Martins et al., 2020).

Để giải quyết các vấn đề này, điều khiển dự báo mô hình (MPC) (Berlin et al., 2021; Pham et al., 2024) và đặc biệt là điều khiển dự báo phi tuyến (NMPC) (Ismael, Almagued, & Abdulla, 2024; Lee, Chung, Kim, & Kim, 2023) đã mở ra một hướng đi mới với khả năng tính toán và điều khiển trong các hệ thống phi tuyến. NMPC sử dụng mô hình dự báo để tối ưu hóa các lệnh điều khiển dựa trên các trạng thái và ràng buộc của hệ thống, giúp cải thiện độ ổn định và tính chính xác trong các điều kiện vận hành phức tạp. Tuy nhiên, việc áp dụng NMPC cho robot di động hai bánh vi sai trong môi trường động với các ràng buộc an toàn và tránh va chạm vẫn chưa được nghiên cứu đầy đủ.

Từ các phân tích bên trên, nghiên cứu này được thực hiện tập trung vào việc phát triển một phương pháp điều khiển dự báo phi tuyến (NMPC) cho robot di động hai bánh vi sai nhằm giải quyết vấn đề bám quỹ đạo ổn định và tránh vật cản động. Đặc biệt, các ràng buộc vận tốc tuyến tính, vận tốc góc và gia tốc vào bài toán tối ưu hóa đã được tích hợp trong nghiên cứu, đồng thời các ràng buộc khoảng cách an toàn đã được đưa ra để đảm bảo di chuyển an toàn trong môi trường thay đổi.

Phương pháp này có tính mới so với các nghiên cứu trước đó nhờ việc kết hợp dự báo phi tuyến với các ràng buộc vật lý chặt chẽ hơn, nâng cao độ chính xác quỹ đạo và cải thiện khả năng tránh vật cản trong điều kiện thực tế. Kết quả ở các thí nghiệm thực tế và việc mô phỏng trong Gazebo đã cho thấy phương pháp NMPC mang lại kết quả vượt trội về độ chính xác và hiệu quả so với phương pháp DWA. Mục tiêu

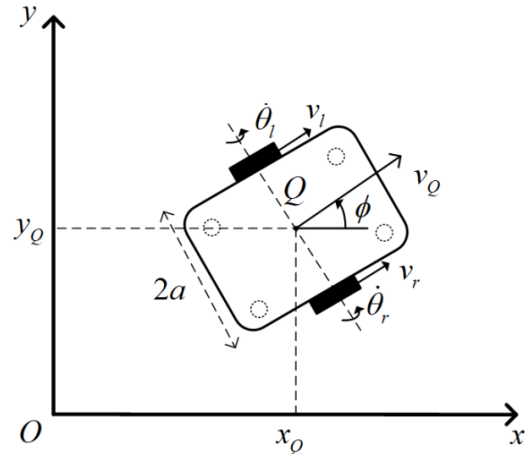
của nghiên cứu là phát triển và đánh giá một phương pháp điều khiển dự báo phi tuyến hiệu quả, có khả năng bám quỹ đạo chính xác và tránh vật cản động trong các tình huống thực tế với robot di động hai bánh vi sai.

2. MÔ HÌNH HÓA VÀ LẬP KẾ HOẠCH QUỸ ĐẠO

2.1. Mô hình hóa hệ thống

2.1.1. Mô hình động học

Robot hai bánh lái vi sai trong Hình 1 có các thông số được chú giải như sau: a là khoảng cách tâm xe đến tâm bánh xe chủ động, r là bán kính bánh xe, v_r là vận tốc dài bánh phải, v_l là vận tốc dài bánh trái, θ_r là vận tốc góc bánh phải, θ_l vận tốc góc bánh trái, ϕ góc đánh lái robot theo trục x , (x_Q, y_Q) tọa độ của robot trong hệ tọa độ Oxy và v_Q là vận tốc của robot.



Hình 1. Robot hai bánh lái vi sai trong hệ tọa độ Oxy

2.1.2. Động học thuận

Phương trình động học thuận của robot 2 bánh lái vi sai (Tzafestas, 2013) trong Hình 1 được trình bày dưới dạng phương trình như sau :

$$\begin{cases} \dot{x}_Q = v_Q \cos \phi \\ \dot{y}_Q = v_Q \sin \phi \\ \dot{\phi} = \omega \end{cases} \quad (1)$$

Trong đó: \dot{x}_Q, \dot{y}_Q và $\dot{\phi}$ là vận tốc dài theo trục x, y và vận tốc đánh lái tương ứng.

Công thức (1) được viết dưới dạng ma trận như sau:

$$\begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_Q \\ \omega \end{bmatrix} \quad (2)$$

2.1.3. Động học nghịch

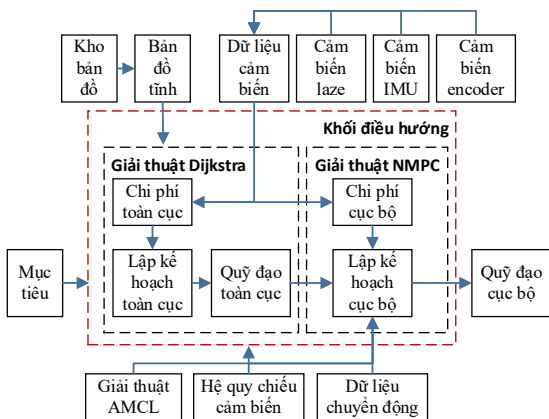
Phương trình động học nghịch của robot 2 bánh lái vi sai (Tzafestas, 2013) trong Hình 1 được trình bày dưới dạng phương trình như sau:

$$\begin{bmatrix} \dot{\theta}_r \\ \dot{\theta}_l \end{bmatrix} = \frac{1}{r} \begin{bmatrix} \cos \phi & \sin \phi & a \\ \cos \phi & \sin \phi & -a \end{bmatrix} \begin{bmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\phi} \end{bmatrix} \quad (3)$$

2.2. Lập kế hoạch quỹ đạo

2.2.1. Tổng quan quy trình điều hướng

Kiến trúc tổng quan của một hệ thống điều hướng trong robot di động đã được mô tả trong Hình 2, bao gồm các thành phần chính như Global Planner, Local Planner, và các giải thuật hỗ trợ như Dijkstra, NMPC và AMCL.

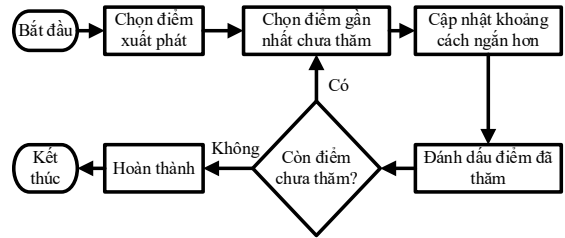


Hình 2. Tổng quan kiến trúc điều hướng

Dữ liệu từ các cảm biến đã được sử dụng trong hệ thống như RPLidar, IMU và Encoder, để xây dựng bản đồ tĩnh và tính toán lộ trình. Các thành phần này được sử dụng phối hợp với nhau trong việc tính toán lộ trình từ điểm xuất phát đến mục tiêu, nhằm đảm bảo tính chính xác và hiệu quả cho quá trình di chuyển của robot.

2.2.2. Quy đạo toàn cục (Dijkstra)

Thuật toán tìm đường quỹ đạo toàn cục trong Hình 3 bắt đầu từ điểm xuất phát và tiếp tục tìm kiếm điểm đích thông qua việc lựa chọn các điểm gần nhất chưa được khám phá.

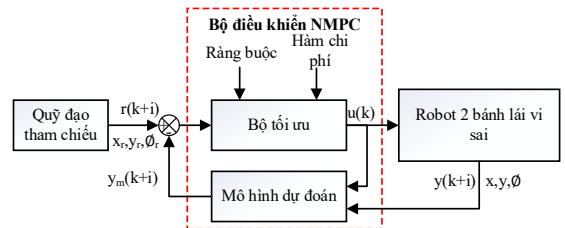


Hình 3. Sơ đồ thuật toán Dijkstra

Quá trình bắt đầu với việc chọn điểm xuất phát, từ đó hệ thống lựa chọn điểm có khoảng cách ngắn nhất mà chưa được khám phá. Sau khi chọn điểm gần nhất, thuật toán cập nhật lại khoảng cách ngắn nhất đến các điểm khác, nhằm tối ưu hóa đường đi. Mỗi lần thực hiện, thuật toán kiểm tra xem còn điểm nào chưa được khám phá và nếu có, tiếp tục chọn điểm gần nhất tiếp theo. Sau khi một điểm được xử lý, nó được đánh dấu là đã khám phá để tránh bị xử lý lại. Quy trình này tiếp tục cho đến khi tất cả các điểm đã được khám phá hoặc khi đường đi đến điểm đích được xác định, kết thúc quá trình tìm kiếm. Thuật toán này giúp đảm bảo việc tìm kiếm đường đi từ điểm xuất phát đến đích một cách tối ưu và hiệu quả.

2.2.3. Quy đạo cục bộ (NMPC)

Giải thuật NMPC là một phương pháp điều khiển robot sử dụng mô hình dự đoán để tối ưu hóa hành vi của robot trong tương lai.



Hình 4. Sơ đồ điều khiển giải thuật NMPC

Trong Hình 4, NMPC tính toán một chuỗi điều khiển tối ưu nhằm giúp robot di chuyển theo một quỹ đạo tham chiếu, đồng thời giảm thiểu sai số giữa vị trí thực tế và mục tiêu mong muốn. Các yếu tố đã được sử dụng cho bộ tối ưu hóa như ràng buộc về vận tốc, vật cản (7) (8), cũng như hàm chi phí (6) để đảm bảo rằng các lệnh điều khiển được tạo ra không vượt quá các giới hạn vật lý và động học của robot, như tốc độ di chuyển tối đa, gia tốc tối đa, và khả năng thay đổi hướng của robot. Điều này giúp đảm bảo rằng robot không bị yêu cầu thực hiện các thao tác vượt quá khả năng của nó, chẳng hạn như di chuyển quá nhanh, thay đổi vận tốc đột ngột hoặc vượt quá độ ổn định trong quá trình điều khiển. Quá trình này được thực hiện lặp đi lặp lại, với mỗi vòng

lập robot cập nhật trạng thái và tính toán điều khiển tối ưu cho các bước tiếp theo, giúp robot duy trì quỹ đạo tham chiếu một cách chính xác. Dưới đây là trình tự thực hiện của giải thuật NMPC:

Xác định mô hình động học của hệ thống (Tzafestas, 2013):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v \cos \phi \\ v \sin \phi \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \phi & 0 \\ \sin \phi & 0 \\ 0 & 1 \end{bmatrix} \mathbf{u} \quad (4)$$

Trong đó $\mathbf{u} = [v \ \omega]^T$ là vector điều khiển.

Rời rạc mô hình động học của hệ thống (Lafmejani & Berman, 2021):

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_k \cos(\phi_k) T \\ \mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{v}_k \sin(\phi_k) T \\ \phi_{k+1} = \phi_k + \omega_k T \end{cases} \quad (5)$$

Trong đó T là thời gian lấy mẫu.

Hàm mục tiêu hệ thống được xác định theo công thức (Lafmejani & Berman, 2021):

$$J = \sum_{k=0}^{N-1} \left((\mathbf{x}_k - \mathbf{x}_f)^T \mathbf{Q} (\mathbf{x}_k - \mathbf{x}_f) + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) \quad (6)$$

Trong đó: $\mathbf{Q} \in \mathbb{R}^{3 \times 3}$ và $\mathbf{R} \in \mathbb{R}^{2 \times 2}$ lần lượt là các ma trận trọng số lỗi vị trí và ma trận trọng số điều khiển. \mathbf{x}_k là trạng thái của hệ thống tại thời điểm k bao gồm các biến số vị trí và tư thế $(x_k, y_k, \phi_k)^T$. \mathbf{x}_f là quỹ đạo tham chiếu được biểu diễn bởi tham số $(x_r, y_r, \phi_r)^T$ được minh họa trong Hình 4. \mathbf{u}_k là tín hiệu điều khiển tại thời điểm k bao gồm các biến số về vận tốc dài và vận tốc góc $[v_k \ \omega_k]^T$. N là số bước dự đoán của hệ thống.

Ràng buộc của hệ thống được xác định theo công thức:

$$\begin{cases} \dot{\mathbf{x}}_k = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{x}_{min} \leq \mathbf{x}_k \leq \mathbf{x}_{max} \\ \mathbf{u}_{min} \leq \mathbf{u}_k \leq \mathbf{u}_{max} \end{cases} \quad (7)$$

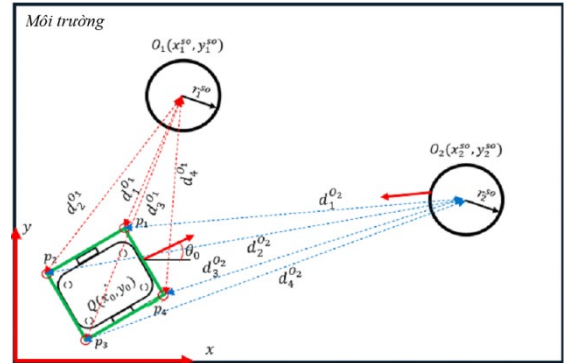
Trong đó $\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ là mô hình động học của hệ thống xe hai bánh lái vi sai.

Dựa vào Hình 5, ràng buộc vật cản được xác định theo công thức:

$$\sqrt{(x_{p_k}(t) - x_{r_i}^{o_i}(t))^2 + (y_{p_k}(t) - y_{r_i}^{o_i}(t))^2} \leq d_{safe} + r_i^{so} \quad (8)$$

Trong đó: d_{safe} là khoảng cách an toàn từ robot đến vật cản, $x_{p_k}(t)$, $y_{p_k}(t)$ là hệ tọa độ thứ k của robot theo thời gian t chuyển động trong hệ tọa độ

xOy , $x_{r_i}^{o_i}(t)$ và $y_{r_i}^{o_i}(t)$ là hệ tọa độ của vật cản thứ i theo thời gian t trong hệ tọa độ xOy , r_i^{so} là bán kính của vật cản thứ i .



Hình 5. Robot di chuyển trong môi trường có vật cản

Để giải bài toán tối ưu (6) thỏa mãn các ràng buộc (7) và (8) công cụ IPOPT được đề xuất Wächter and Biegler (2006) và đã được sử dụng để tìm ra nghiệm tối ưu cho hàm chi phí. Luật điều khiển tối ưu được xác định theo biểu thức:

$$\mathbf{u}^* = \underset{\mathbf{u}_k}{\operatorname{arg\,min}} J \quad (9)$$

Trong đó \mathbf{u}^* là luật điều khiển tối ưu cho hệ thống với dãy tín hiệu điều khiển u_0^*, \dots, u_{N-1}^* .

Ngoài ra, trong phần này, cửa sổ dự đoán được thiết lập với số bước $N = 20$, cùng khoảng thời gian giữa các bước dự đoán là 0,3 giây. Cấu hình này nhằm cân bằng giữa độ chi tiết của quỹ đạo điều khiển và khả năng xử lý của bộ giải, đảm bảo hệ thống phản hồi hiệu quả trong quá trình điều hướng.

Thực hiện điều khiển tín hiệu u_0^* đầu tiên và sau đó cập nhật trạng thái và lặp lại quy trình. Cuối cùng việc chạy mô phỏng và tinh chỉnh các ma trận trọng số Q và R đã được tiến hành sao cho đảm bảo độ mượt mà quỹ đạo và đảm bảo tránh vật cản an toàn.

2.2.4. Phương pháp xác định vị trí (AMCL)

AMCL được xem như một bước cải tiến quan trọng so với phương pháp Monte Carlo Localization (MCL) truyền thống. Đây là hệ thống định vị hoạt động trong không gian hai chiều (2D), dựa trên bộ lọc hạt nhằm xác định vị trí và hướng di chuyển (tư thế) của robot. Trong đó, mỗi hạt được coi là một giả thuyết về tư thế của robot và được gán một trọng số phản ánh mức độ phù hợp của giả thuyết này với dữ liệu thu thập từ các cảm biến.

Điểm ưu việt của AMCL nằm ở khả năng tự động điều chỉnh số lượng hạt trong quá trình tính

toán. Điều này giúp giảm thiểu chi phí tính toán nhưng vẫn đảm bảo độ chính xác cao trong định vị. Nhờ vậy, AMCL không chỉ tăng cường hiệu suất xử lý thời gian thực mà còn trở thành một trong những kỹ thuật cốt lõi, được sử dụng phổ biến trong các hệ thống robot di động hiện đại. Thuật toán được thực hiện theo trình tự các bước sau (Chung & Lin, 2021; Lattitia, 2022):

Khởi tạo số lượng hạt $x_t^{[m]}$ và trọng số $w_t^{[m]}$ với tập hợp M hạt và khởi tạo tập hợp các hạt ban đầu là rỗng:

$$\bar{X}_t = X_t = \emptyset \quad (10)$$

Trạng thái của từng hạt được dự đoán dựa trên mô hình chuyển động của robot:

$$x_t^{[m]} = \text{motion_update}(u_t, x_{t-1}^{[m]}) \quad (11)$$

Trọng số của mỗi hạt được tính toán dựa trên thông tin cảm biến theo công thức:

$$w_t^{[m]} = \text{sensor_update}(z_t, x_t^{[m]}) \quad (12)$$

Việc tái lấy mẫu đã được tiến hành để loại bỏ những hạt có trọng số thấp và tăng cường những hạt có trọng số cao, với công thức:

$$\bar{X}_t = X_t + \langle x_t^{[m]}; w_t^{[m]} \rangle \quad (13)$$

Quá trình trên được thực hiện liên tục theo thời gian thực, dựa vào dữ liệu cảm biến cập nhật cho đến khi thuật toán hội tụ và xác định chính xác vị trí của robot trên bản đồ.

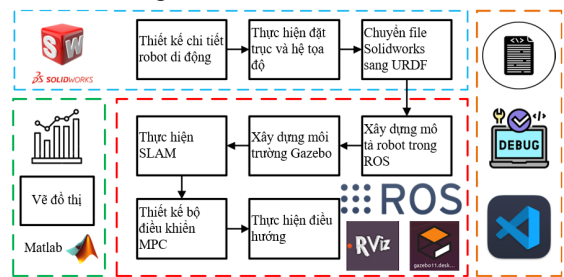
3. KẾT QUẢ VÀ THẢO LUẬN

Trong nghiên cứu này, bộ điều khiển dự báo phi tuyến (NMPC) được kiểm chứng thông qua cả mô phỏng trên Gazebo và thực nghiệm trên mô hình robot hai bánh lái vi sai. Ở môi trường mô phỏng, robot được xây dựng trên SolidWorks và triển khai trong Gazebo để khảo sát các tình huống thay đổi tầm nhìn cảm biến, vận tốc, tải trọng cũng như khi di chuyển trong môi trường có vật cản tĩnh và động. Để có thể chứng minh tính hiệu quả của phương pháp đề xuất. Một số mô phỏng điều hướng tránh vật cho robot trên gazebo và thực nghiệm trên mô hình thực tế được triển khai. Tiếp đó, mô hình thực nghiệm được chế tạo và triển khai với các cảm biến Lidar, IMU, Encoder và bộ điều khiển nhúng, tiến hành thử nghiệm trong các kịch bản thay đổi tải trọng và môi trường động. Các kết quả thực nghiệm tiếp tục khẳng định tính ưu việt của NMPC khi quỹ đạo di chuyển mượt mà hơn, sai số vị trí và góc nhỏ hơn so với DWA, đồng thời cho thấy khả năng thích ứng tốt với sự thay đổi môi trường. Tuy nhiên, kết

quả nghiên cứu cũng chỉ ra những hạn chế nhất định như độ trễ khi robot vận hành ở tốc độ cao, sự phụ thuộc vào độ chính xác của mô hình hệ thống và yêu cầu phần cứng mạnh để xử lý trong thời gian thực.

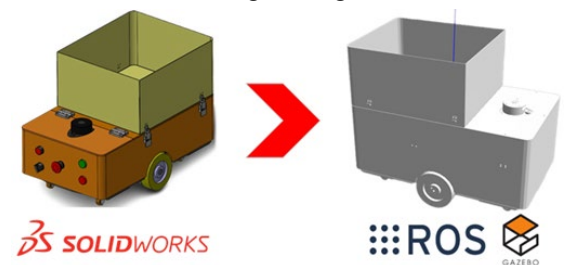
3.1. Kiểm chứng trên mô phỏng Gazebo

Quy trình tổng quan minh họa Hình 6 trình bày từ khâu thiết kế robot trên SolidWorks, chuyển đổi sang URDF và xây dựng mô hình trong ROS, đến việc tạo môi trường mô phỏng bằng Gazebo, thực hiện SLAM, thiết kế bộ điều khiển NMPC và kiểm chứng điều hướng. Các công cụ MATLAB, RViz và Visual Studio Code hỗ trợ việc phân tích, trực quan hóa và gỡ lỗi, đảm bảo quá trình mô phỏng toàn diện và có hệ thống.



Hình 6. Tổng quan quy trình mô phỏng

Bên cạnh đó, Hình 7 minh họa việc chuyển đổi mô hình robot được thiết kế trên SolidWorks sang định dạng URDF và tích hợp vào ROS - Gazebo. Quá trình này giúp tái hiện mô hình cơ khí trong môi trường mô phỏng, tạo điều kiện kiểm chứng các thuật toán SLAM, NMPC và điều hướng trước khi triển khai trên hệ thống thực nghiệm.



Hình 7. Robot chuyển từ SolidWorks sang ROS

Thêm vào đó, Bảng 1 thể hiện các tham số động lực học của bánh xe trong URDF, gồm hệ số ma sát, độ cứng và hệ số giảm chấn tiếp xúc. Các thông số này giúp mô hình mô phỏng phản ánh chính xác đặc tính vật lý, nâng cao độ tin cậy khi kiểm chứng điều khiển.

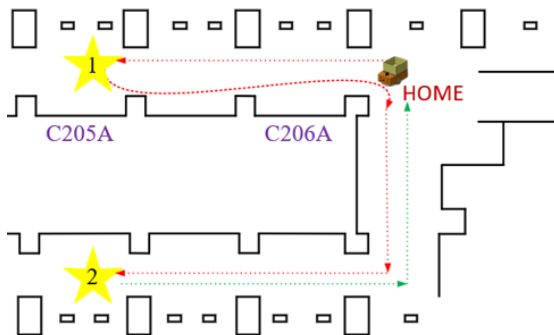
Hơn thế nữa, Hình 8 minh họa lộ trình mô phỏng robot trong các trường hợp: không có vật cản, có vật cản tĩnh và có vật cản động. Với các tiêu chí đánh

giá, bao gồm: thời gian, quãng đường, vận tốc đáp ứng và sai số vị trí, sai số góc.

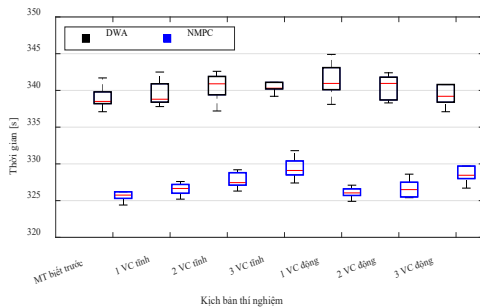
Bảng 1. Bảng tham số bánh xe mô phỏng

Thông số	Trong URDF	Giá trị
Hướng ma sát	Fdir1	(1 0 0)
Ma sát chính	mu1	0,9
Ma sát phụ	mu2	0,9
Độ cứng tiếp xúc	kp	5000000 N/m
Hệ số giảm chấn tiếp xúc	kd	10 Ns/m

Kết quả so sánh từ 10 lần lấy mẫu được trình bày dưới dạng biểu đồ hộp (Tukey, 1977) cho thấy bộ điều khiển dự báo (NMPC) vượt trội so với DWA ở hầu hết các tiêu chí đánh giá.

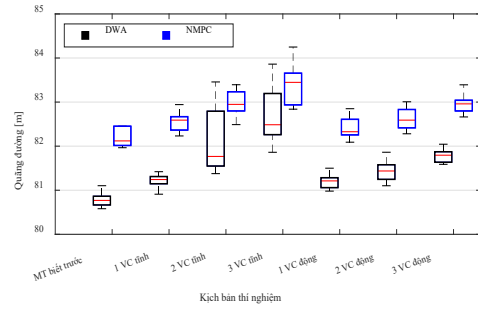


Hình 8. Kế hoạch chuyển động robot trong môi trường mô phỏng

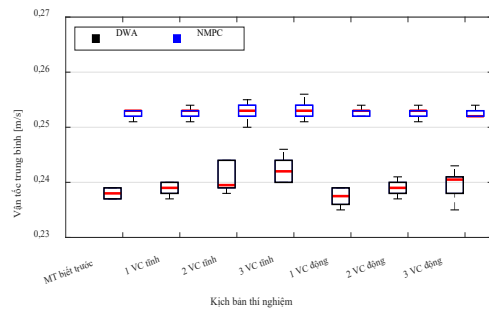


Hình 9. So sánh phân bố thời gian giữa giải thuật NMPC với giải thuật DWA trong môi trường thay đổi vật cản

Thời gian hoàn thành nhiệm vụ của NMPC Hình 9 luôn ngắn hơn, với trung vị thấp hơn 10 - 15 giây và độ phân tán nhỏ, thể hiện tính ổn định cao. Quãng đường di chuyển Hình 10 của NMPC thường dài hơn, nhưng dữ liệu tập trung quanh trung vị, đảm bảo độ tin cậy và an toàn trong môi trường có vật cản.

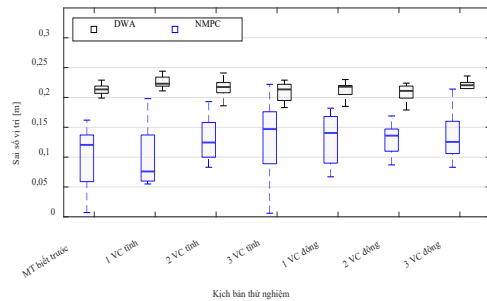


Hình 10. So sánh phân bố quãng đường giữa giải thuật NMPC với giải thuật DWA trong môi trường thay đổi vật cản

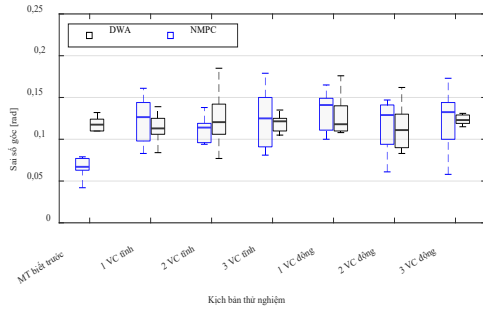


Hình 11. So sánh phân bố vận tốc trung bình giữa giải thuật NMPC với giải thuật DWA trong môi trường thay đổi vật cản

Vận tốc trung bình của NMPC được thể hiện tại Hình 11 ổn định hơn, dao động trong khoảng 0,252 - 0,254 m/s, cao hơn DWA (0,238 - 0,242 m/s). Sai số vị trí của NMPC Hình 12 chỉ ở mức 0,10 - 0,16 m, thấp hơn đáng kể so với DWA, trong khi sai số góc được minh họa ở Hình 13 cũng nhỏ và ổn định hơn.



Hình 12. Đánh giá sai số vị trí giữa giải thuật NMPC và DWA khi không mang tải

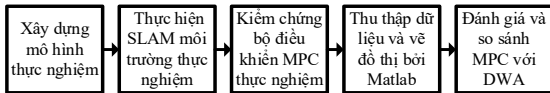


Hình 13. Đánh giá sai số góc giữa giải thuật NMPC và DWA khi không mang tải

Nhìn chung, NMPC không chỉ duy trì độ chính xác và tốc độ tốt mà còn thể hiện tính thích nghi và tin cậy cao hơn khi điều hướng trong môi trường động phức tạp. Những kết quả này khẳng định tính khả thi của phương pháp và tạo nền tảng vững chắc để tiếp tục kiểm chứng trên mô hình thực nghiệm.

3.2. Kiểm chứng trong môi trường thực nghiệm

Sau khi hoàn tất giai đoạn mô phỏng, nghiên cứu được triển khai trong môi trường thực nghiệm nhằm đánh giá tính khả thi của bộ điều khiển NMPC. Quy trình thực nghiệm được trình bày tại Hình 14 bao gồm: xây dựng mô hình robot, thực hiện SLAM trong môi trường thực, kiểm chứng bộ điều khiển MPC thực nghiệm, thu thập dữ liệu và vẽ đồ thị bởi Matlab, đồng thời việc so sánh kết quả giữa NMPC và DWA cũng đã được tiến hành. Cách tiếp cận này cho phép kiểm chứng toàn diện, bảo đảm tính khách quan và độ tin cậy trước khi ứng dụng vào thực tiễn.



Hình 14. Tổng quan quy trình thực nghiệm

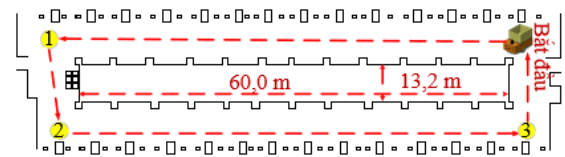
Bên cạnh đó, mô hình robot lái vi sai được triển khai trong thực nghiệm có cấu trúc và hình dáng tương đồng với phiên bản thiết kế trong mô phỏng, như minh họa ở Hình 15.



Hình 15. Robot lái vi sai thực nghiệm

Hơn thế nữa, bên trong robot hai bánh lái vi sai, hệ thống nhúng Jetson Orin NX 16GB với hiệu suất AI 100 TOPS và GPU 1024 lõi Ampere đảm nhận việc xử lý dữ liệu và tính toán trong thời gian thực. Bộ vi xử lý 8 nhân ARM Cortex-A78AE (tối đa 2GHz) và bộ nhớ 16GB LPDDR5 (102,4GB/s) giúp xử lý nhanh chóng các tín hiệu từ cảm biến như LiDAR AIM8 và IMU HWT901B. Hệ thống hỗ trợ kết nối nhanh qua PCIe Gen4 và USB 3.2, đồng thời các tác vụ điều khiển và tránh vật cản được xử lý đồng thời mà không gặp phải độ trễ đáng kể, mang lại hiệu suất ổn định trong môi trường động.

Môi trường thực nghiệm Hình 16 được bố trí tại hành lang với kích thước 60,0 m × 13,2 m. Tại đây, robot thực hiện các kịch bản điều hướng để kiểm chứng và so sánh hiệu quả giữa MPC và DWA trong điều kiện thực tế.

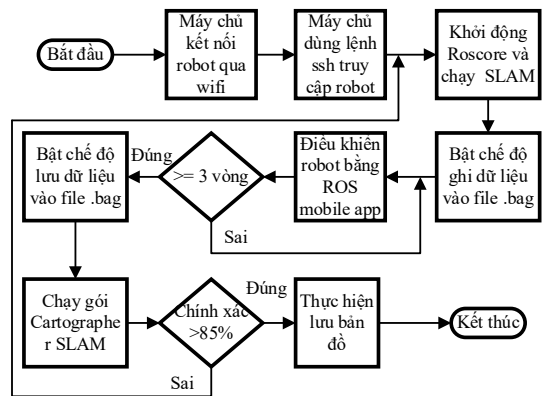


Hình 16. Môi trường thực nghiệm

Quá trình xây dựng bản đồ thực nghiệm được tiến hành từ tập dữ liệu cảm biến được thu thập trước đó dưới dạng tệp có đuôi “.bag” bằng giải thuật Cartographer tại tầng 2, khu C, Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh. Quy trình thực hiện được minh họa tại Hình 17. Bản đồ thu được sau đó được đánh giá chất lượng thông qua công thức sau (Norzam et al., 2019):

$$Accuracy(\%) = \frac{x}{y} \times 100 \quad (14)$$

Trong đó: x là chu vi của bản đồ thu thập được và y là chu vi của bản đồ thực tế.



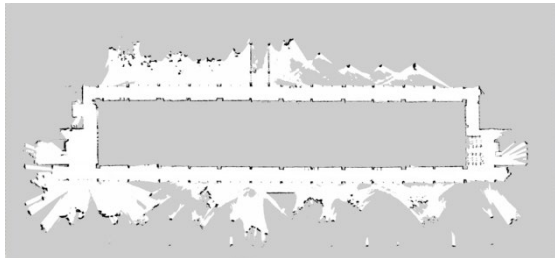
Hình 17. Quy trình xây dựng bản đồ thực nghiệm

Trong Bảng 2, ba bản đồ từ Cartographer SLAM được so sánh theo vận tốc, số vòng quét, thời gian, góc nghiêng và độ chính xác. Bản đồ 1 hoàn thành nhanh nhất nhưng độ phủ hạn chế và góc nghiêng lớn, gây sai lệch quỹ đạo. Bản đồ 2 ổn định hơn song độ chính xác thấp nhất. Bản đồ 3 mất nhiều thời gian nhất do quét ba vòng, nhưng đạt độ phủ không gian tốt, sai số nhỏ và độ chính xác cao nhất (97,09%).

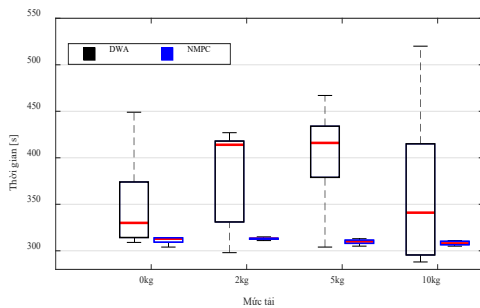
Bảng 2. Bảng so sánh đánh giá các trường hợp Cartographer SLAM

Tiêu chí đánh giá	Bản đồ 1	Bản đồ 2	Bản đồ 3
Vận tốc dài (m/s)	0,2	0,2	0,2
Vận tốc góc (rad/s)	0,1	0,2	0,2
Thời gian (giây)	882,7	1884,4	2497,1
Số vòng (vòng)	1	2	3
Góc nghiêng (độ)	4,21	0,77	0,82
Chiều dài (mét)	59,2	59,2	59,3
Chiều rộng (mét)	10,6	10,6	10,6
Chu vi (mét)	139,6	139,5	139,8
Độ chính xác (%)	96,95	96,88	97,09

Do đó, bản đồ 3 minh họa Hình 18 được đánh giá tối ưu và lựa chọn cho giai đoạn điều hướng.



Hình 18. Bản đồ sau khi thực hiện SLAM

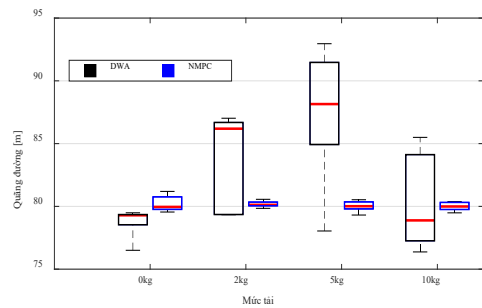


Hình 19. So sánh phân bố thời gian giữa giải thuật NMPC với giải thuật DWA trong môi trường tĩnh thay đổi tải

Kết quả thực nghiệm trong môi trường tĩnh thay đổi tải sau 5 lần lấy mẫu cho mỗi trường hợp bao gồm các tải trọng 0 kg, 2 kg, 5 kg và 10 kg cho thấy

NMPC đạt hiệu quả điều hướng ổn định hơn so với DWA.

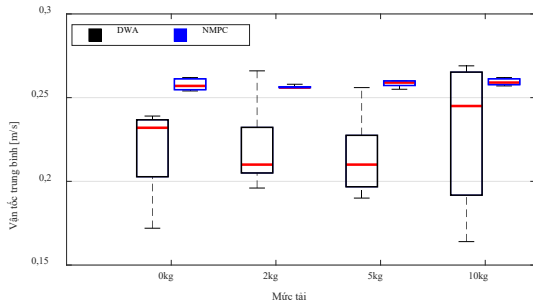
Trước hết, trong Hình 19 dễ dàng nhận thấy thuật toán DWA dao động mạnh về thời gian hoàn thành khi thay đổi tải, với phương sai và độ lệch chuẩn cao (ở tải 5 kg, $\sigma \approx 60$ s) và IQR rộng (tới 73 s), phản ánh kết quả thiếu ổn định. Ngược lại, NMPC ổn định vượt trội, phương sai và độ lệch chuẩn hầu hết dưới 5 s, IQR gần như bằng 0, các giá trị tập trung sát trung vị. Việc so sánh trung bình cho thấy DWA thường chậm hơn và phân tán lớn, trong khi NMPC vừa nhanh hơn vừa ổn định. Như vậy, khi tải tăng, DWA kéo dài và biến động nhiều, còn NMPC vẫn duy trì ổn định cả về tốc độ lẫn độ lặp lại.



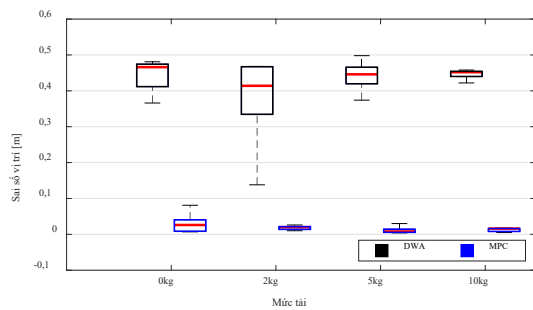
Hình 20. So sánh phân bố quãng đường giữa giải thuật NMPC với giải thuật DWA trong môi trường tĩnh thay đổi tải

Tiếp theo, Hình 20 cho thấy DWA dao động mạnh về quãng đường khi thay đổi tải. Ở tải 2 kg, độ lệch chuẩn gần 4 m và IQR 7,2 m; ở 5 kg, IQR vẫn cao tới 3,75 m, phản ánh sự phân tán lớn và thiếu ổn định. Ngược lại, NMPC duy trì ổn định vượt trội với độ lệch chuẩn và IQR đều dưới 1 m, các giá trị tập trung quanh trung vị và có tính lặp lại cao. Về quãng đường trung bình, DWA thường dài hơn ($\approx 89,3$ m ở 5 kg), trong khi NMPC ổn định quanh 80 m bất kể tải trọng. Như vậy, NMPC ổn định và hiệu quả hơn, còn DWA có xu hướng tạo lộ trình dài và biến động khi tải tăng.

Bên cạnh đó, Hình 21 cho thấy NMPC luôn đạt vận tốc cao và ổn định hơn DWA ở mọi mức tải. Ở tải 0 kg, NMPC đạt 0,258 m/s với σ chỉ 0,0036 m/s, trong khi DWA chỉ 0,218 m/s và σ tới 0,028 m/s. Tại 2 kg, DWA tiếp tục dao động mạnh ($\sigma \approx 0,027$ m/s, IQR = 0,013 m/s), còn NMPC gần như tuyệt đối ổn định ($\sigma < 0,001$ m/s, IQR = 0). Ở 10 kg, DWA biến động lớn với nhiều giá trị lệch trung vị, trong khi NMPC vẫn giữ ổn định. Như vậy, NMPC vừa đạt vận tốc trung bình cao hơn, vừa ổn định hơn, còn DWA giảm vận tốc và dao động mạnh khi tải tăng.

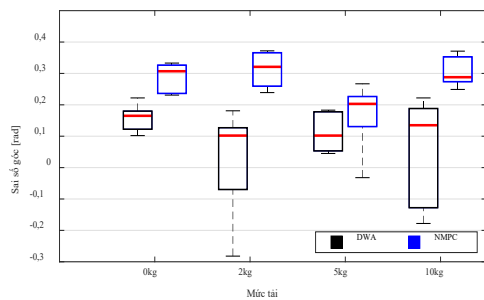


Hình 21. So sánh phân bố vận tốc trung bình giữa giải thuật NMPC với giải thuật DWA trong môi trường tĩnh thay đổi tải



Hình 22. Đánh giá sai số vị trí giữa giải thuật NMPC và DWA trong môi trường tĩnh thay đổi tải

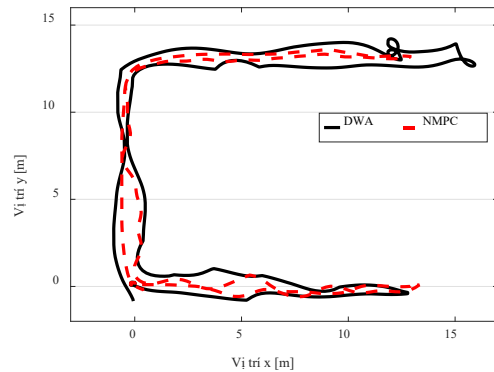
Hơn nữa, Hình 22 cho thấy DWA có sai số vị trí trung bình lớn (0,37 – 0,45 m) ở mọi mức tải, với đồ thị hộp phân bố rộng, độ lệch chuẩn cao và tính ổn định kém, đặc biệt ở tải 2 kg (0,377 m). Ngược lại, NMPC vượt trội về độ chính xác khi sai số trung bình chỉ 0,011 – 0,030 m, các giá trị gần như trùng nhau, đồ thị hộp hẹp và tập trung sát trung vị, chứng tỏ độ lặp lại cao. Như vậy, DWA duy trì sai số lớn và biến động mạnh, còn NMPC luôn ổn định với sai số cực nhỏ, cho thấy khả năng điều khiển chính xác và thích ứng vượt trội.



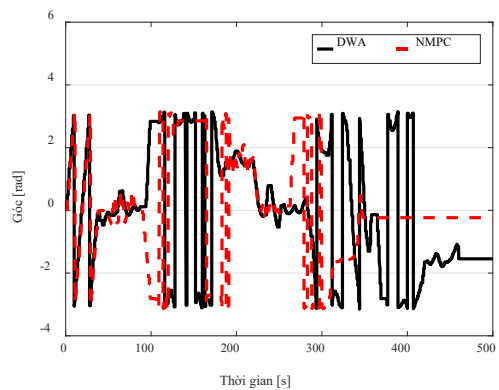
Hình 23. Đánh giá sai số góc giữa giải thuật NMPC và DWA trong môi trường tĩnh thay đổi tải

Cuối cùng, Hình 23 cho thấy DWA cho sai số góc nhỏ và ổn định hơn NMPC. Ở tải 0 kg, sai số trung bình của DWA là 0,157 rad, thấp hơn nhiều so với NMPC (0,287 rad); ở 2 kg, DWA chỉ còn 0,022 rad trong khi NMPC vẫn cao (~ 0,312 rad). Với tải 5 - 10 kg, DWA dao động 0,049 – 0,113 rad, còn NMPC cao hơn (0,167 – 0,307 rad). Đồ thị hộp cũng phản ánh sự khác biệt: DWA phân bố hẹp và tập trung quanh trung vị, trong khi NMPC phân tán rộng và sai số lớn. Như vậy, về sai số góc tại đích 1, DWA vượt trội hơn NMPC, trái ngược với các chỉ khác (quãng đường, vận tốc, sai số vị trí) nơi NMPC thường ưu thế.

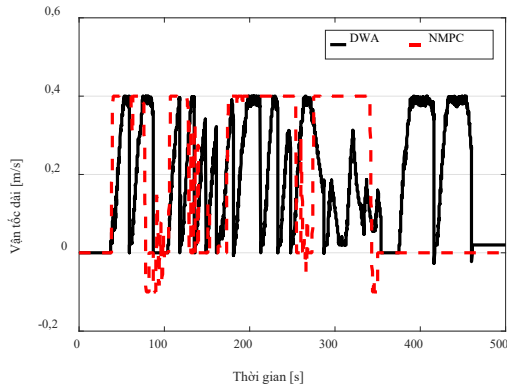
Kết quả thực nghiệm trong môi trường động cho thấy NMPC đạt hiệu quả điều hướng ổn định hơn so với DWA.



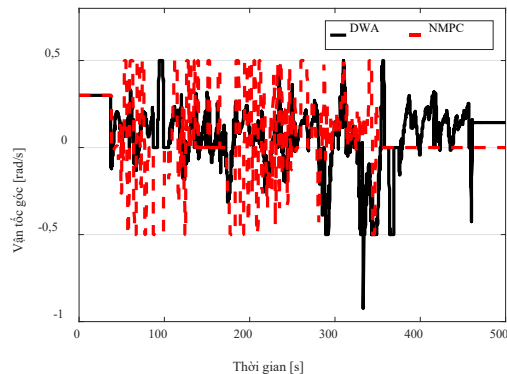
Hình 24. Quỹ đạo di chuyển của robot trong môi trường động



Hình 25. Đồ thị góc θ trong môi trường động



Hình 26. Đồ thị vận tốc dài trong môi trường động



Hình 27. Đồ thị vận tốc góc trong môi trường động

Quỹ đạo di chuyển của NMPC Hình 24 bám sát đường đi, ít dao động và sai lệch. Góc quay θ của NMPC Hình 25 biến thiên nhỏ hơn, giúp robot duy trì hướng ổn định. Vận tốc tuyến tính tại Hình 26 và

vận tốc góc tại Hình 27 của NMPC cũng ít biến động, phân bố đều quanh giá trị trung bình, trong khi DWA dao động mạnh và thiếu ổn định. Nhìn chung, NMPC chứng tỏ khả năng duy trì quỹ đạo mượt mà, kiểm soát chính xác và thích ứng tốt hơn trong điều kiện môi trường động phức tạp.

4. KẾT LUẬN

Kết quả nghiên cứu này giúp trình bày và kiểm chứng phương pháp điều khiển dự báo phi tuyến (NMPC) cho robot di động hai bánh lái vi sai, giải quyết vấn đề bám quỹ đạo ổn định và tránh vật cản trong môi trường phức tạp. Phương pháp NMPC đã chứng minh hiệu quả vượt trội so với các phương pháp truyền thống, đặc biệt trong việc xử lý các ràng buộc về vận tốc, gia tốc và khoảng cách an toàn. Kết quả mô phỏng trên phần mềm Gazebo và thí nghiệm thực tế cho thấy phương pháp này giúp robot duy trì quỹ đạo chính xác, tránh va chạm và thích ứng với thay đổi môi trường. Tuy nhiên, việc tối ưu hóa trong thời gian thực đòi hỏi phần cứng mạnh mẽ và có thể gặp độ trễ khi robot vận hành nhanh. Độ chính xác của mô hình động học cũng ảnh hưởng đến hiệu quả trong môi trường thay đổi nhanh. Do đó, việc cải tiến thuật toán NMPC cần được thực hiện để giảm chi phí tính toán và nâng cao hiệu quả thực tế.

Trong tương lai, việc kết hợp NMPC với các phương pháp học máy để cải thiện khả năng tự học và tự thích nghi của robot là cần thiết. Ngoài ra, việc nghiên cứu các phương pháp tối ưu hóa nhanh hơn, giảm yêu cầu phần cứng và cải thiện thời gian phản hồi cũng rất quan trọng. Việc tích hợp công nghệ mới như mạng cảm biến không dây, trí tuệ nhân tạo và big data sẽ mở ra cơ hội mới cho ứng dụng robot di động trong môi trường phức tạp.

TÀI LIỆU THAM KHẢO

Berlin, J., Hess, G., Karlsson, A., Ljungbergh, W., Zhang, Z., Åkesson, K., & Götvald, P.-L. (2021). *Trajectory generation for mobile robots in a dynamic environment using nonlinear model predictive control*. Paper presented at the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE). <https://doi.org/10.1109/CASE49439.2021.9551644>

Chung, M.-A., & Lin, C.-W. (2021). An improved localization of mobile robotic system based on AMCL algorithm. *IEEE Sensors Journal*, 22(1), 900-908. <https://doi.org/10.1109/JSEN.2021.3126605>

Ismael, O. Y., Almaged, M., & Abdulla, A. I. (2024). Nonlinear model predictive control-based collision avoidance for mobile robot. *Journal of*

Robotics and Control (JRC), 5(1), 142-151. <https://doi.org/10.18196/jrc.v5i1.20615>

Konar, A., Chakraborty, I. G., Singh, S. J., Jain, L. C., & Nagar, A. K. (2013). A deterministic improved Q-learning for path planning of a mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(5), 1141-1153.

<https://doi.org/10.1109/TSMCA.2012.2227719>

Lafmejani, A. S., & Berman, S. (2021). Nonlinear MPC for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots. *Robotics and Autonomous Systems*, 141, 103774.

<https://doi.org/10.1016/j.robot.2021.103774>

- Lauttia, T. (2022). *Adaptive Monte Carlo Localization in ROS*. Tampere University Tampere, Finland.
- Lee, C., Chung, D., Kim, J., & Kim, J. (2023). Nonlinear model predictive control with obstacle avoidance constraints for autonomous navigation in a canal environment. *IEEE/ASME Transactions on Mechatronics*, 29(3), 1985-1996.
<https://doi.org/10.1109/TMECH.2023.3312672>
- Li, X., Liu, F., Liu, J., & Liang, S. (2017). Obstacle avoidance for mobile robot based on improved dynamic window approach. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(2), 666-676.
<https://doi.org/10.3906/elk-1504-194>
- Malone, N., Chiang, H.-T., Lesser, K., Oishi, M., & Tapia, L. (2017). Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field. *IEEE Transactions on robotics*, 33(5), 1124-1138.
<https://doi.org/10.1109/TRO.2017.2705034>
- Martins, O., Adekunle, A., Adejuyigbe, S., Adeyemi, O., & Arowolo, M. (2020). Wheeled Mobile Robot Path Planning and Path Tracking Controller Algorithms: A Review. *Journal of Engineering Science & Technology Review*, 13(3).
<https://doi.org/10.25103/jestr.133.17>
- Norzam, W., Hawari, H., & Kamarudin, K. (2019). *Analysis of mobile robot indoor mapping using GMapping based SLAM with different parameter*. Paper presented at the IOP Conference Series: Materials Science and Engineering.
<https://doi.org/10.1088/1757-899X/705/1/012037>
- Pham, Q. A. H., Le, Q. T., Le, M. H., & Tran, D. T. (2024). *A Model Predictive Control for Dynamic Obstacle Avoidance of a Mobile Robot Based on ROS*. Paper presented at the 2024 7th International Conference on Green Technology and Sustainable Development (GTSD).
- Tukey, J. W. (1977). *Exploratory data analysis*. Reading/Addison-Wesley.
- Tzafestas, S. G. (2013). *Introduction to mobile robot control*. Elsevier.
<https://doi.org/10.1016/B978-0-12-417049-0.00008-0>
- Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106(1), 25-57.
<https://doi.org/10.1007/s10107-004-0559-y>