

TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ



ĐỒ ÁN THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN
QUÁ TRÌNH SẢN XUẤT
ĐỀ TÀI
NGHIÊN CỨU, THIẾT KẾ VÀ THỰC NGHIỆM BỘ ĐIỀU
KHIỂN THÍCH NGHI BÁM QUỲ ĐẠO CHO ROBOT

Giảng viên hướng dẫn : ThS. Hồ Sỹ Phương

Sinh viên thực hiện : Trần Phương Nam

Lớp : 62K2

Ngành : Kỹ thuật điều khiển và tự động hóa

NGHỆ AN, 2025

ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN

Họ và tên sinh viên: Trần Phương Nam – 215752021610076

Ngành: Kỹ thuật điều khiển và tự động hóa

Đề tài: Nghiên cứu, thiết kế và thực nghiệm bộ điều khiển bám quỹ đạo cho robot.

1. Nhiệm vụ:

- Tìm hiểu và xác định nội dung đồ án.
- Nghiên cứu mô hình toán học, động học, động lực học.
- Thiết kế bộ điều khiển, mô phỏng kiểm chứng bằng Matlab Simulink
- Lập trình, tiến hành thực nghiệm cho mô hình.
- Đánh giá kết quả thực nghiệm.
- Kết luận và hướng phát triển.

2. Ngày giao đồ án: Ngày 1 Tháng 9 năm 2024

3. Ngày hoàn thành đồ án: Ngày 10 Tháng 01 Năm 2025

Người hướng dẫn: ThS. Hồ Sỹ Phương

Nghệ An, Ngày ... Tháng ... Năm 202..

TRƯỞNG BỘ MÔN

NGƯỜI HƯỚNG DẪN

BẢN TỰ ĐÁNH GIÁ

Nhóm trưởng: Trần Phương Nam

Đề tài: Nghiên cứu, thiết kế và thực nghiệm bộ điều khiển bám quỹ đạo cho robot.

Phản đánh giá

| TT | Họ và tên MSSV | Nội dung thực hiện | Kết quả | Đánh giá(%) | Ghi chú |
|----|------------------------------------|--|---------------|----------------|----------------|
| 1 | Trần Phương Nam 215752021610076 | - Nghiên cứu mô phỏng - Lập trình cho Robot -Thực nghiệm và tối ưu thuật toán | Hoàn thành | 100% | Nhóm trưởng |

Nghệ An, Ngày Tháng 01 Năm 2025

NHÓM TRƯỞNG

NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

Họ và tên sinh viên: Trần Phương Nam – 215752021610076

Ngành: Kỹ thuật điều khiển và tự động hóa

Đề tài: Nghiên cứu, thiết kế và thực nghiệm bộ điều khiển bám quỹ đạo cho robot.

Nhận xét chung:

.....
.....

Phần đánh giá

| TT | Họ và tên MSSV | Nội dung thực hiện | Kết quả | Đánh giá(%) | Ghi chú |
|----|------------------------------------|---|------------|----------------|----------------|
| 1 | Trần Phương Nam 215752021610076 | - Nghiên cứu mô phỏng - Lập trình cho Robot - Thực nghiệm và tối ưu thuật toán | | | Nhóm trưởng |

- Ý thức thực hiện:

- Nội dung thực hiện:.....

- Hình thức trình bày:.....

Tổng hợp kết quả(điểm bằng số và bằng chữ):

+Điểm chuyên cần:.....

+Điểm giữa kỳ:.....

Nghệ An, Ngày ... Tháng 01 Năm 2025

NGƯỜI HƯỚNG DẪN

MỤC LỤC

| | |
|--|-----------|
| TÓM TẮT | 1 |
| CHƯƠNG 1. MÔ HÌNH HOÁ ROBOT | 3 |
| 1.1 Mô hình toán học | 3 |
| 1.2 Mô hình động học | 4 |
| 1.3 Mô hình động lực học | 4 |
| CHƯƠNG 2. THIẾT KẾ BỘ ĐIỀU KHIỂN..... | 7 |
| 2.1 Bộ điều khiển Backstepping..... | 7 |
| 2.2 Bộ điều khiển thích nghi dựa trên mạng Neural..... | 8 |
| 2.3 Bộ điều khiển PID..... | 13 |
| CHƯƠNG 3. MÔ PHỎNG VÀ THỰC NGHIỆM | 15 |
| 3.1 Kết quả mô phỏng..... | 15 |
| 3.2 Kết quả thực nghiệm | 16 |
| CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN | 22 |
| 4.1 Kết luận | 22 |
| 4.2 Hướng phát triển | 22 |
| TÀI LIỆU THAM KHẢO..... | 24 |

DANH MỤC HÌNH ẢNH

| | |
|---|----|
| Hình 1.1 Cấu trúc robot di động..... | 3 |
| Hình 2.1 Sơ đồ cấu trúc bộ điều khiển đề xuất | 7 |
| Hình 2.2 Cấu trúc các lớp của mạng nơ ron | 10 |
| Hình 2.3 Sơ đồ khối bộ điều khiển pid | 13 |
| Hình 2.4 Chu kỳ dao động pcr | 14 |
| Hình 3.1 Quỹ đạo mô phỏng | 15 |
| Hình 3.2 Sai số mô phỏng | 15 |
| Hình 3.3 Các tham số được tinh chỉnh | 16 |
| Hình 3.4 Sơ đồ khối | 17 |
| Hình 3.5 Bản vẽ mô hình | 17 |
| Hình 3.6 Mô hình thực nghiệm | 18 |
| Hình 3.7 Tham số điều khiển được huấn luyện và chỉnh định theo điều kiện thực nghiệm | 20 |
| Hình 3.8 Quỹ đạo thực nghiệm | 20 |
| Hình 3.9 Sai lệch đo được trong quá trình theo dõi quỹ đạo đặt | 21 |

TÓM TẮT

Robot di động 2 bánh vi sai (DDMR) đã trở thành một nền tảng phổ biến trong các ứng dụng công nghiệp và dịch vụ nhờ tính linh hoạt, dễ thiết kế và triển khai trong nhiều môi trường khác nhau. Tuy nhiên, việc điều khiển chính xác quỹ đạo của robot trong thực tế vẫn là một bài toán khó do sự xuất hiện của nhiễu và các yếu tố môi trường thay đổi liên tục [1][2]. Các phương pháp điều khiển truyền thống như PID hay điều khiển trượt (SMC) thường được sử dụng nhờ tính hiệu quả trong các tình huống tiêu chuẩn, nhưng chúng lại phụ thuộc nhiều vào mô hình động lực học chính xác. Điều này dẫn đến hạn chế trong việc xử lý các yếu tố bất định và biến đổi không dự đoán trước, đặc biệt khi hệ thống hoạt động trong môi trường phức tạp [3][4].

Để vượt qua những thách thức này, mạng nơ-ron nhân tạo (ANN) đã được triển khai trong các bộ điều khiển robot như một công cụ mạnh mẽ, nhờ khả năng học tập và thích nghi với các điều kiện thay đổi. ANN có thể giảm sự phụ thuộc vào các mô hình toán học phức tạp, cho phép robot tự động điều chỉnh các tham số điều khiển, từ đó cải thiện hiệu suất và độ chính xác trong việc bám quỹ đạo [5][6]. Nhiều nghiên cứu đã chứng minh ANN có khả năng ước lượng chính xác các tham số hệ thống, đồng thời xử lý hiệu quả các yếu tố bất định và nhiễu động trong môi trường thực tế [7][8]. Tuy nhiên, đa số các nghiên cứu này chỉ dừng lại ở mức mô phỏng, chưa được kiểm chứng đầy đủ trên các hệ thống thực tế [9].

Trong đồ án này, nhóm thực hiện phát triển và kiểm chứng một bộ điều khiển thích nghi dựa trên mạng nơ-ron dành cho DDMR. Bộ điều khiển được thiết kế để tự học và điều chỉnh tham số động học chưa biết, từ đó tối ưu hóa khả năng bám quỹ đạo trong môi trường có nhiễu và yếu tố biến động. Đặc biệt, bộ điều khiển không chỉ giúp giảm thiểu sai số mà còn đảm bảo độ ổn định cao ngay cả trong điều kiện hoạt động khắc nghiệt [10]. Kết quả thực nghiệm đã chứng minh bộ điều khiển đạt hiệu suất vượt trội so với các phương pháp truyền thống, với khả năng duy trì tính chính xác và ổn định của hệ thống trong nhiều tình huống khác nhau [11].

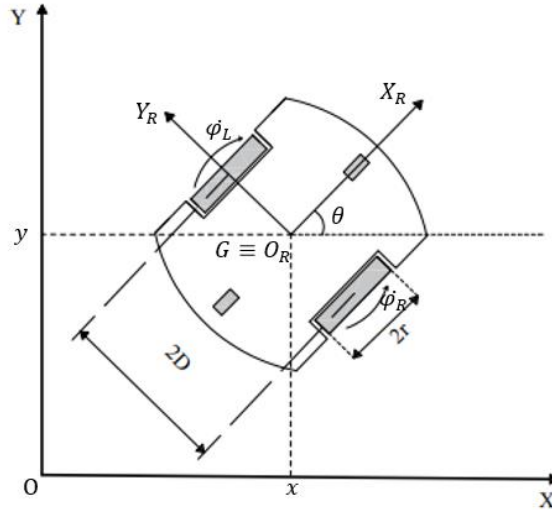
Nội dung đồ án được tổ chức thành 4 phần chính. Phần 1 trình bày chi tiết quá trình mô hình hóa robot. Phần 2 đi sâu vào mô tả bộ điều khiển dựa trên mạng nơ-ron, tập trung vào cách thức tự động chỉnh định tham số. Phần 3 phân tích kết quả thực nghiệm, minh chứng hiệu quả của bộ điều khiển trong điều kiện thực tế. Cuối cùng, phần 4 kết

luyện và đề xuất các hướng phát triển trong tương lai, nhấn mạnh tiềm năng ứng dụng của đồ án trong các hệ thống tự động hóa khác [12].

CHƯƠNG 1. MÔ HÌNH HOÁ ROBOT

1.1 Mô hình toán học

Cấu trúc của robot di động được thiết kế có bốn bánh xe. Hai bánh xe ở giữa là 2 bánh chủ động, được điều khiển bởi hai động cơ DC. Một bánh xe phía trước và sau thân robot là 2 bánh thụ động. Cấu trúc robot được thể hiện trong *Hình 1.1*. Điểm G là trọng tâm robot, 2 động cơ được sắp xếp đồng trục.



Hình 1.1 Cấu trúc robot di động

Hệ toạ độ tại điểm G $\{O_R X_R Y_R\}$ gắn trên thân robot được gọi là hệ toạ độ robot hoặc hệ toạ cục bộ vì hệ toạ độ này được dịch chuyển theo chuyển động của robot di động. Hệ toạ độ còn lại $\{OXY\}$ được gọi là hệ toạ độ quán tính hay hệ toạ độ cố định, được sử dụng khi làm việc trên vị trí của robot. Hai hệ toạ độ này lần lượt được biểu diễn như sau:

$$\begin{cases} q_R = [x_r \ y_r \ \theta_r]^T \\ q = [x \ y \ \theta]^T \end{cases} \quad (1)$$

Một ma trận biến đổi $Rot(\theta)$ được sử dụng để chuyển đổi giữa hệ toạ độ cố định và hệ toạ độ robot:

$$\dot{q}_R = Rot(\theta) \dot{q} \Rightarrow \dot{q} = Rot(\theta)^{-1} \dot{q}_R \quad (2)$$

Trong đó:

$$Rot(\theta)^{-1} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

1.2 Mô hình động học

Động học của robot liên quan đến mối quan hệ giữa vị trí, vận tốc, gia tốc và các cấu trúc vật lý của nó. Bằng cách thay đổi vận tốc của hai bánh xe, quỹ đạo của robot có thể thay đổi. $\dot{\phi}_R, \dot{\phi}_L$ lần lượt là vận tốc quay của bên phải và bên trái. Ta có:

$$\begin{aligned} V_R &= r\dot{\phi}_R \\ V_L &= r\dot{\phi}_L \end{aligned} \quad (4)$$

V_R và V_L là vận tốc tuyến tính bánh xe phải và trái. Với bất kỳ trường hợp nào, v và ω có thể được tính như sau:

$$\begin{aligned} v &= \frac{(V_R + V_L)}{2} \\ \omega &= \frac{(V_R - V_L)}{2D} \end{aligned} \quad (5)$$

Từ (3) và (4), mối quan hệ vận tốc giữa hai toạ độ được biểu diễn như sau:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{(V_R + V_L)}{2} \cos \theta \\ \frac{(V_R + V_L)}{2} \sin \theta \\ \frac{(V_R - V_L)}{2D} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (6)$$

Mô hình động học thuận của robot có thể được miêu tả như sau:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = S(q)V(t) \quad (7)$$

1.3 Mô hình động lực học

Phân tích động lực học của robot có thể được định nghĩa là kiểm tra mối quan hệ giữa độ lớn mô-men xoắn hoặc độ lớn lực tác dụng lên bánh xe bởi bộ truyền động, vị trí, vận tốc và gia tốc của robot theo thời gian. Sự khác biệt giữa mô hình động lực học và

mô hình động học là mô hình động học chỉ xem xét chuyển động với các quan hệ hình học chi phối hệ thống mà không xem xét các lực tác động đến chuyển động.

Theo [12], mô hình động lực học tính được như sau:

$$\begin{aligned} M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d \\ = B(q)\tau - A^T(q)\lambda \end{aligned} \quad (8)$$

q, \dot{q} và \ddot{q} lần lượt chỉ định các vector vị trí, vận tốc và gia tốc tương ứng. $M(q)$ là ma trận quán tính xác định dương đối xứng, $V(q, \dot{q})$ là ma trận hướng tâm và Coriolis, $G(q)$ là lực hấp dẫn vector, τ_d biểu thị các nhiễu hoặc nhiễu không xác định giới hạn, bao gồm cả động lực học phi cấu trúc không được mô hình hóa, $B(q)$ là ma trận biến đổi đầu vào, τ là vector mô-men xoắn đầu vào, $A^T(q)$ là ma trận được liên kết với các ràng buộc và λ là vector của các lực ràng buộc. Bỏ qua lực hấp dẫn $G(q)$, ma sát của bánh xe được truyền vào hệ có sai số được xác định trong mô hình động học nên $F(q)$ bị bỏ qua, giá trị nhiễu được cộng vào tín hiệu điều khiển của cơ cấu chấp hành nên ta bỏ qua trong mô hình động. m là khối lượng và I biểu thị mô men quán tính của robot, ta có:

$$\begin{aligned} M(q) &= \begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{pmatrix}, G(q) = 0, F(q) = 0 \\ B(q) &= \frac{1}{r} \begin{bmatrix} \cos \theta & \cos \theta \\ \sin \theta & \sin \theta \\ D & -D \end{bmatrix}, \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \\ V(q, \dot{q}) &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, A^T(q)\lambda = \begin{bmatrix} -\sin \theta \\ \cos \theta \\ 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \end{aligned} \quad (9)$$

Phương trình động lực học của robot được viết lại như sau:

$$M(q)\ddot{q} = B(q)\tau - A^T(q)\lambda \quad (10)$$

Ma trận $S(q)$ được sử dụng để rút gọn. Ma trận $S(q)$ là ma trận động học tịnh tiến được sửa đổi liên quan đến khoảng cách giữa trọng tâm của robot và trục bánh xe, như đã thấy trong phương trình động học tịnh tiến:

$$\dot{q} = S(q)\eta \quad (11)$$

Ma trận gia tốc thu được bằng cách lấy đạo hàm thời gian phương trình (11) như sau:

$$\ddot{q} = \dot{S}(q)\dot{\eta} + S(q)\ddot{\eta} \quad (12)$$

Sử dụng phương trình (10) và (12), mô hình robot được sửa như sau:

$$M(q)S(q)\dot{\eta} + M(q)\dot{S}(q)\dot{\eta} = B(q)\tau - A^\tau(q)\lambda \quad (13)$$

Nhân cả 2 vế cho ma trận biến đổi $S^T(q)$, ta có:

$$\begin{aligned} S^T(q)M(q)S(q)\dot{\eta} + S^T(q)M(q)\dot{S}(q)\dot{\eta} \\ = S^T(q)B(q)\tau - S^T(q)A^\tau(q)\lambda \end{aligned} \quad (14)$$

Ma trận $S(q)$ nằm trong không gian rỗng của ma trận ràng buộc động học $A(q)$, có nghĩa là $S^T(q)A^\tau(q) = 0$. Vì vậy, phương trình (14) trở thành:

$$\overline{M}(q)\dot{\eta} + \overline{V}(q, \dot{q})\eta = \overline{B}(q)\tau \quad (15)$$

Các thuật ngữ mô hình động trong phương trình (15) có thể được trình bày như sau:

$$\begin{aligned} \overline{M}(q) &= S^T(q)M(q)S(q) = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \\ \overline{V}(q, \dot{q}) &= S^T(q)M(q)\dot{S}(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \overline{B}(q) &= S^T(q)B(q) = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ D & -D \end{bmatrix} \end{aligned} \quad (16)$$

Sử dụng vận tốc trong phương trình (5), phương trình (15) có thể được chuyển đổi thành dạng thay thế. Cấu trúc này trong phương trình (17) là mô hình phi tuyến.

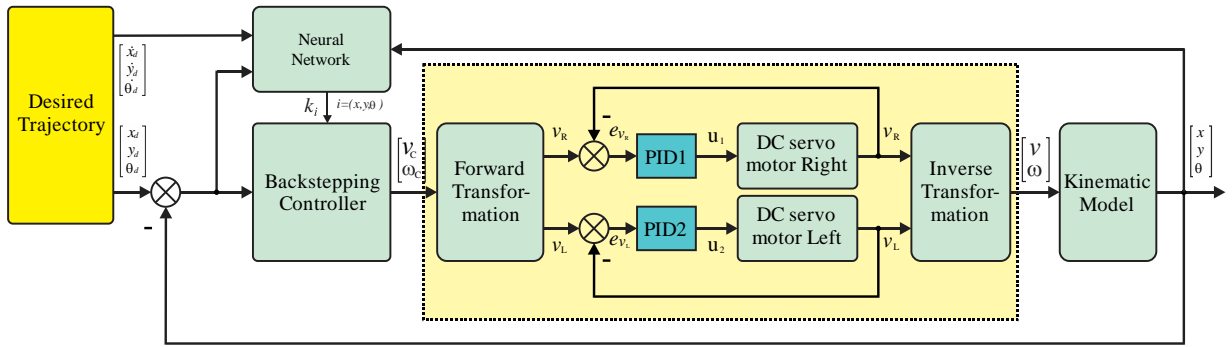
$$\begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 1 \\ D & -D \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \Rightarrow \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{rm} & \frac{1}{rm} \\ \frac{D}{rI} & \frac{-D}{rI} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (17)$$

Với mô hình phi tuyến, vận tốc tuyến tính và vận tốc góc thu được bằng cách sử dụng phương trình (17).

CHƯƠNG 2. THIẾT KẾ BỘ ĐIỀU KHIỂN

Phương pháp điều khiển theo quỹ đạo thường được áp dụng trong các môi trường đã biết và dựa trên động học của robot. Tuy nhiên, khi các yếu tố động lực như quán tính, khối lượng hay vị trí khối lượng thay đổi, cần phải xem xét đến động lực học của robot. Mục tiêu chính là giúp robot bám sát quỹ đạo đề ra, nhưng sự phức tạp của hệ thống cùng các yếu tố bất định, nhiễu loạn hay lỗi cảm biến có thể gây ra sai lệch trong chuyển động. Để giải quyết những thách thức này, bộ điều khiển được đề xuất có khả năng học hỏi từ động lực học chưa biết, giúp robot bám theo quỹ đạo với độ chính xác cao trong môi trường mô phỏng.

Cấu trúc bộ điều khiển đề xuất được thể hiện trên *Hình 2.1* với sự tham gia của bộ điều khiển động lực học với khả năng thích nghi nhờ sử dụng kiến trúc mạng nơ-ron thể hiện trong *Hình 2.2*.



Hình 2.1 Sơ đồ cấu trúc bộ điều khiển đề xuất

2.1 Bộ điều khiển Backstepping

Với $q = [x \ y \ \theta]^T$ và $q_R = [x_r \ y_r \ \theta_r]^T$ lần lượt là vị trí trong hệ tọa độ cố định và hệ tọa độ robot. Sự khác biệt giữa q và q_R cho các giá trị sai lệch về vị trí của robot trong hệ tọa độ chuẩn. Phương trình sai lệch bám vị trí của robot được chỉ ra[18]:

$$e_q = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{pmatrix} \quad (18)$$

Đạo hàm cả 2 vế ta có:

$$\dot{e}_q = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_r \cos \theta_e \\ -\omega x_e + v_r \sin \theta_e \\ \omega_r - \omega \end{bmatrix} \quad (19)$$

Dựa trên cách tiếp cận được phát triển trong [18] luật điều khiển động học được áp dụng như sau:

$$\begin{pmatrix} v_c \\ \omega_c \end{pmatrix} = \begin{pmatrix} v_r \cos \theta_e + K_x x_e \\ \omega_r + v_r (K_y y_e + K_\theta \sin \theta_e) \end{pmatrix} \quad (20)$$

K_x, K_y và K_θ là các giá trị dương. Thông thường, để thiết kế các tham số này đòi hỏi quá trình nhận dạng hệ thống với các thuật toán phức tạp. Ở đây nhóm đề xuất một mạng nơ ron để điều chỉnh các tham số liên tục giúp robot có khả năng thích ứng tốt hơn trước các nhiễu loạn và các nhân tố ảnh hưởng chưa biết.

2.2 Bộ điều khiển thích nghi dựa trên mạng Neural

Bộ điều khiển thích nghi dựa trên mạng nơ-ron theo phương pháp backstepping được đề xuất cung cấp các tham số và hệ số thích nghi thay đổi theo quỹ đạo tham chiếu. Cấu trúc điều khiển điều chỉnh các hệ số của bộ điều khiển dựa trên động học để tối thiểu hóa hàm chi phí sau:

$$J = \frac{1}{2} \sum g_x x_e^2 + g_y y_e^2 + g_\theta \theta_e^2 \quad (21)$$

Ở đây, g_x, g_y , và g_θ là các hệ số mạng nơ-ron. Các giá trị sai số được lấy từ phương trình (18). K_x, K_y, K_θ được xem là một phần của hàm chi phí ở trên và được tối ưu hóa cũng như cập nhật dựa trên phương pháp gradient descent. Các hệ số điều khiển động học được biểu diễn bởi vector α . Đạo hàm riêng của hàm chi phí đối với α được tính như sau:

$$\frac{\partial J}{\partial \alpha} = g_x x_e \frac{\partial x_e}{\partial \alpha} + g_y y_e \frac{\partial y_e}{\partial \alpha} + g_\theta \theta_e \frac{\partial \theta_e}{\partial \alpha} = e_p^T g \frac{\partial e_p}{\partial \alpha} \quad (22)$$

với $\alpha = [K_x \ K_y \ K_\theta]$

Dạng ma trận của giá trị g được thể hiện như sau:

$$g = \begin{bmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_\theta \end{bmatrix} \quad (23)$$

Trong Phương trình (22), nếu e_p được thay thế bởi Phương trình (14):

$$\frac{\partial J}{\partial \alpha} = g e_p^T \frac{\partial (T_e(q_r - q_c))}{\partial \alpha} = -e_p^T \times g \times T_e \times \frac{\partial q_c}{\partial \alpha} \quad (24)$$

Vì vector q_r là vị trí tham chiếu bao gồm các giá trị cố định, đạo hàm của nó bằng 0. Theo quy tắc chuỗi, phương trình sau có thể được viết:

$$\frac{\partial q_c}{\partial \alpha} = \frac{\partial q_c}{\partial v_c} \times \frac{\partial v_c}{\partial \alpha} \quad (25)$$

Sau khi có được Phương trình (25), biểu thức đạo hàm mong muốn trong Phương trình (26) được viết như sau:

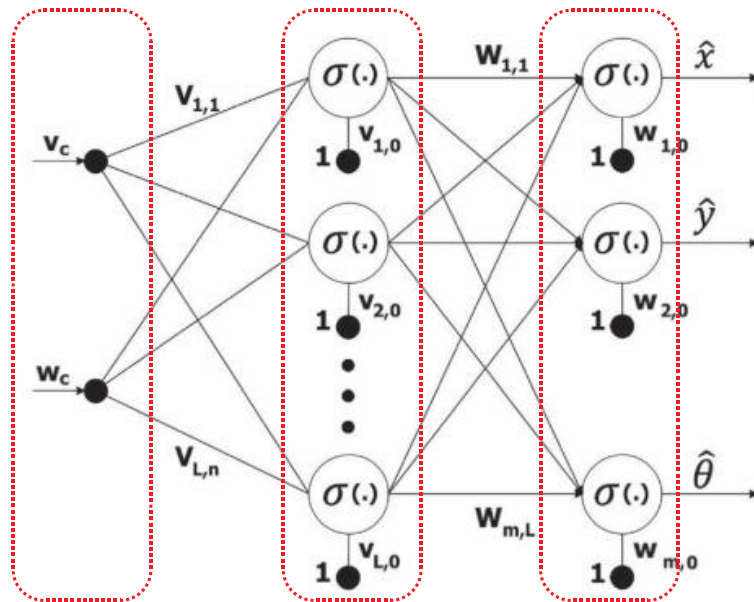
$$\frac{\partial q_c}{\partial \alpha} = \begin{bmatrix} Jac_{v11} & Jac_{v12} \\ Jac_{v21} & Jac_{v22} \\ Jac_{v31} & Jac_{v32} \end{bmatrix} \times \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin e_\theta \end{bmatrix} \quad (26)$$

Đạo hàm của hàm chi phí đối với các hệ số điều khiển $\frac{\partial J}{\partial \alpha}$ được viết lại như sau:

$$\frac{\partial J}{\partial \alpha} = \left[\frac{\partial J}{\partial K_x} \frac{\partial J}{\partial K_y} \frac{\partial J}{\partial K_\theta} \right] = -e_p^T \times g \times T_e \times Jac_v \times \frac{\partial v_c}{\partial \alpha} \quad (27)$$

Biểu diễn đạo hàm của hàm chi phí theo các hệ số điều khiển ở dạng ma trận được viết như sau:

$$\begin{aligned} \frac{\partial J}{\partial \alpha} &= \left[\frac{\partial J}{\partial K_x} \frac{\partial J}{\partial K_y} \frac{\partial J}{\partial K_\theta} \right] = -[x_e \ y_e \ \theta_e] \times \begin{bmatrix} g_x & 0 & 0 \\ 0 & g_y & 0 \\ 0 & 0 & g_\theta \end{bmatrix} \times \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} Jac_{v11} & Jac_{v12} \\ Jac_{v21} & Jac_{v22} \\ Jac_{v31} & Jac_{v32} \end{bmatrix} \times \begin{bmatrix} e_x & 0 & 0 \\ 0 & v_r e_y & v_r \sin e_\theta \end{bmatrix} \end{aligned} \quad (28)$$



Hình 2.2 Cấu trúc các lớp của mạng nơ-ron

Thuật toán tính ma trận Jacobian bằng mô hình mạng nơ-ron (NN): Các giá trị trọng số ban đầu nên phù hợp nhất với mô hình. Trọng số được điều chỉnh bằng cách huấn luyện mô hình sử dụng thuật toán lan truyền ngược (back propagation). Cần thiết phải có các giá trị không chính xác để áp dụng thuật toán lan truyền ngược. Do đó, thuật toán lan truyền tiến (forward propagation) được áp dụng trước tiên. Thuật toán này được sử dụng để lấy dữ liệu dự đoán do NN tạo ra. Vì đây là dữ liệu dự đoán, nên tồn tại sự khác biệt giữa giá trị ước tính và giá trị thực. Sự khác biệt này được sử dụng để xác định giá trị sai số, lan truyền từ lớp đầu ra đến lớp đầu vào trên NN, dẫn đến "lan truyền ngược" với các bước đạo hàm khác nhau.

Mô hình NN với hai lớp neuron được mô tả trong Hình 2.2.

Lớp đầu tiên của mô hình NN nhận L neuron, và lớp thứ hai nhận m neuron. Các thuật ngữ sau được sử dụng để mô tả các tham số khác nhau của mạng nơ-ron như thể hiện trong Hình 3:

- Vận tốc điều khiển động học v_c và w_c là đầu vào cho NN.
- $\hat{x}, \hat{y}, \hat{\theta}$ là các đầu ra của NN.
- N_i và N_o lần lượt là số lượng đầu vào và đầu ra của mạng nơ-ron.
- N_h là số lượng neuron trong lớp ẩn.

- ζ và δ là các hệ số học và tốc độ động lượng.
- σ là hàm kích hoạt (activation function).
- Whi là trọng số từ lớp ẩn đến lớp đầu vào.
- Who là trọng số từ lớp đầu ra đến lớp ẩn.

Theo mô hình mạng nơ-ron được định nghĩa ở trên, mối quan hệ giữa đầu vào và đầu ra của mạng được mô tả như sau:

$$y_i = \sigma \left(\sum_{l=1}^L W_{oh_{il}} \sigma \left(\sum_{j=1}^n Whi_{lj} x_j + v_{l0} \right) + w_{i0} \right) \quad (29)$$

Trong đó $i = 1, 2, \dots, m$ và $1, 2, \dots, L$.

Hàm kích hoạt σ được chọn là hàm sigmoid, được biểu diễn trong phương trình (30):

$$\sigma = \frac{1}{1 + e^{-\rho}} \quad (30)$$

Thuật toán lan truyền ngược cần đạo hàm của hàm kích hoạt. Đạo hàm của hàm sigmoid đối với ρ được viết như sau:

$$\frac{d}{d\rho} \sigma(\rho) = \frac{1}{(1 + e^{-\rho})^2} = \sigma(\rho)(1 - \sigma(\rho)) \quad (31)$$

Biểu thức ρ trong phương trình (30) là đại diện cho tích của trọng số và giá trị neuron đến lớp mong muốn.

Thuật toán lan truyền ngược (Back Propagation) là một thuật toán điều chỉnh trọng số dựa trên phương pháp gradient descent. Cập nhật trọng số được thực hiện như sau:

$$\begin{cases} W_{il}(k+1) = W_{il}(k) - \zeta \frac{\partial E(k)}{\partial W_{il}} \\ V_{lj}(k+1) = V_{lj}(k) - \zeta \frac{\partial E(k)}{\partial V_{lj}} \end{cases} \quad (32)$$

Tín hiệu đầu ra được cung cấp như một tham chiếu cho mạng nơ-ron, và sự khác biệt giữa tín hiệu đầu ra này và đầu ra (y_i) do mạng nơ-ron tạo ra được coi là lỗi (**error**). Các trọng số được thay đổi theo chu kỳ cho đến khi lỗi được tối thiểu hóa bởi mạng.

Lỗi được tính như sau:

$$e_i(k) = Y_i(k) - y_i(k) \quad (33)$$

Trong đó:

$$Y_i(k) = [x_A \quad y_A \quad \alpha_A]^T \quad (34)$$

Sử dụng hàm lỗi, hàm chi phí được tính như sau:

$$E(k) = \frac{1}{2} e_i(k)^2 \quad (35)$$

Các gradient yêu cầu của hàm chi phí $E(k)$ theo trọng số được xác định bằng cách sử dụng quy tắc dây chuyền (chain rule).

$$\begin{aligned} \frac{\partial E}{\partial W_{il}} &= [e_i \times (-1) \times \sigma'(u_i^2)] \times z_l = -z_l \sigma'(u_i^2) e_i \\ \frac{\partial E}{\partial V_{il}} &= \left[\sum_{i=1}^m (-\sigma'(u_i^2) e_i) \times W_{il} \times \sigma'(u_l^1) \right] \times x_j = -x_j \sigma'(u_l^1) \sum_{i=1}^m (\sigma'(u_i^2) e_i + W_{il}) \end{aligned} \quad (36)$$

Trong đó, z_l là đầu ra hàm sigmoid của nơ ron l ở lớp ẩn. Ma trận Jacobian được xác định như sau:

$$Jac_v = \begin{bmatrix} \frac{\partial x}{\partial v_c} & \frac{\partial x}{\partial \omega_c} \\ \frac{\partial y}{\partial v_c} & \frac{\partial y}{\partial \omega_c} \\ \frac{\partial \theta}{\partial v_c} & \frac{\partial \theta}{\partial \omega_c} \end{bmatrix} = \begin{bmatrix} \frac{\hat{x}(t) - \hat{x}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{x}(t) - \hat{x}(t-1)}{\omega_c(t) - \omega_c(t-1)} \\ \frac{\hat{y}(t) - \hat{y}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{y}(t) - \hat{y}(t-1)}{\omega_c(t) - \omega_c(t-1)} \\ \frac{\hat{\alpha}(t) - \hat{\alpha}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{\alpha}(t) - \hat{\alpha}(t-1)}{\omega_c(t) - \omega_c(t-1)} \end{bmatrix} \quad (37)$$

Thông qua mạng nơ ron, bộ điều khiển động học trong phương trình (20) thay đổi các giá trị điều chỉnh và thích nghi để làm cho hàm chi phí bằng 0 theo phương pháp gradient descent như sau:

$$\begin{cases} k_x(t) = k_x(t-1) + \Delta k_x \\ k_y(t) = k_y(t-1) + \Delta k_y \\ k_\theta(t) = k_\theta(t-1) + \Delta k_\theta \end{cases} \quad (38)$$

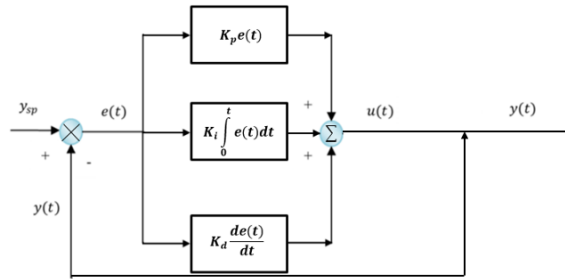
Trong đó, sự thay đổi của các hệ số được tính toán theo các phương trình sau:

$$\begin{aligned}
\Delta k_x &= -\zeta_{K_x} \frac{\partial J}{\partial K_x} \\
\Delta k_y &= -\zeta_{K_y} \frac{\partial J}{\partial K_y} \\
\Delta k_\theta &= -\zeta_{K_\theta} \frac{\partial J}{\partial K_\theta}
\end{aligned} \tag{39}$$

Các hệ số $\zeta_{K_x}, \zeta_{K_y}, \zeta_{K_\theta}$ là các hệ số học của thuật toán gradient descent.

2.3 Bộ điều khiển PID

Phương pháp điều khiển vòng kín thông thường được điều khiển bằng bộ điều khiển PID hay còn gọi là giải thuật PID. PID là tên gọi của bộ điều khiển gồm 3 thành phần Proportional (P) là tỉ lệ, Integral (I) là tích phân và Derivative (D) là vi phân. Bộ điều khiển PID hiệu chỉnh sai số giữa giá trị đo được và giá trị mong muốn bằng cách tính toán ra một tín hiệu điều khiển để giữ cho sai lệch ở mức nhỏ nhất có thể được. Bộ điều khiển này có 3 tham số K_p, K_i và K_d để điều chỉnh sai số của bộ điều khiển.



Hình 2.3 Sơ đồ khối bộ điều khiển PID

Phương trình bộ điều khiển PID được sử dụng:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \tag{40}$$

Sai lệch $e(t)$ tại thời điểm t : độ lệch giữa giá trị mong muốn và giá trị đo được.

$$e(t) = y(t) - y_{sp} \tag{41}$$

Trong đó :

- + y_{sp} : giá trị mong muốn.
- + $y(t)$: giá trị đo được tại thời điểm t .

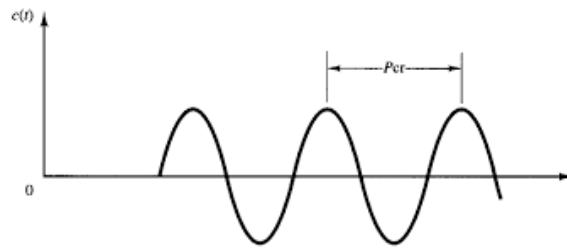
Theo Hình 2.1 bộ điều khiển PID được sử dụng để điều chỉnh cho vận tốc tuyến

tính và vận tốc góc của robot bám với giá trị mong muốn dựa vào độ sai lệch. Từ đó điều khiển 2 động cơ gắn với 2 bánh chủ động của robot. Độ sai lệch được tính toán trong bộ điều khiển PID là độ lệch giữa vận tốc đặt với vận tốc thực tế.

Có nhiều phương pháp xác định tham số của bộ điều khiển PID, trong bài báo này nhóm tác giả sử dụng phương pháp Ziegler-Nichols thứ 2.

Các bước xác định thông số PID cho hệ thống theo phương pháp Ziegler–Nichols thứ 2:

- + Đầu tiên đặt $K_i = 0$ và $K_d = 0$.
- + Chỉ sử dụng tác động khuếch đại, tăng K_p từ 0 tới một giá trị tới hạn K_{cr} , tại đó đầu ra của hệ thống có dạng dao động điều hoà.
- + Xác định K_p và chu kỳ dao động P_{cr} bằng thực nghiệm.



Hình 2.4 Chu kỳ dao động P_{cr}

Xác định các tham số K_p , K_i và K_d của bộ điều khiển theo bảng sau:

| Type of Controller | K_p | K_i | K_d |
|--------------------|--------------|--------------|---------------|
| P | $0.5K_{cr}$ | 0 | 0 |
| PI | $0.45K_{cr}$ | $0.85P_{cr}$ | 0 |
| PID | $0.6K_{cr}$ | $0.5P_{cr}$ | $0.125P_{cr}$ |

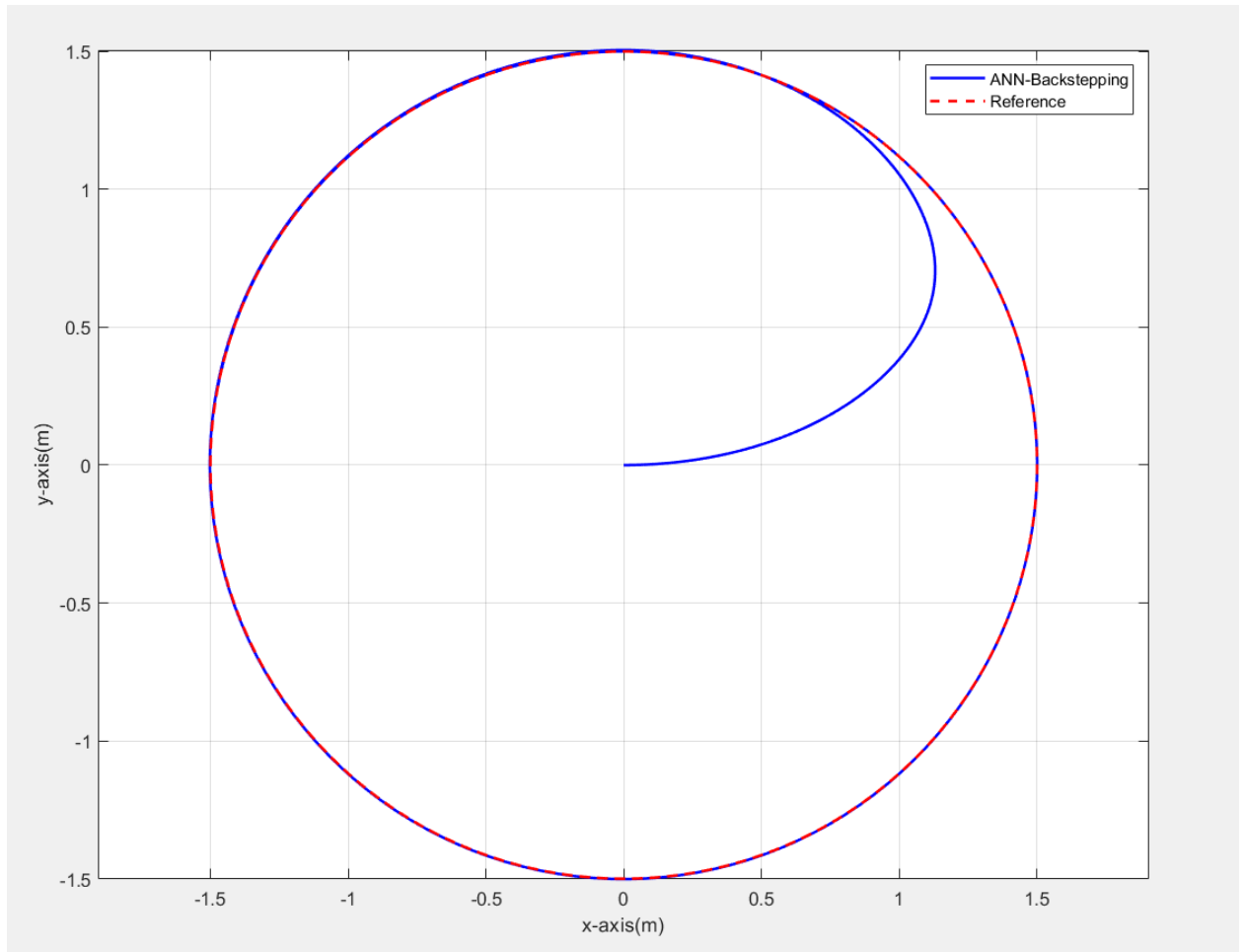
Bảng 2.1 Phương pháp Ziegler–Nichols

Tính toán K_i và K_d của hệ thống dựa vào P_{cr} , từ đó tìm được tham số của bộ điều khiển PID. Sau quá trình thực nghiệm, các giá trị của bộ điều khiển tìm được như sau:

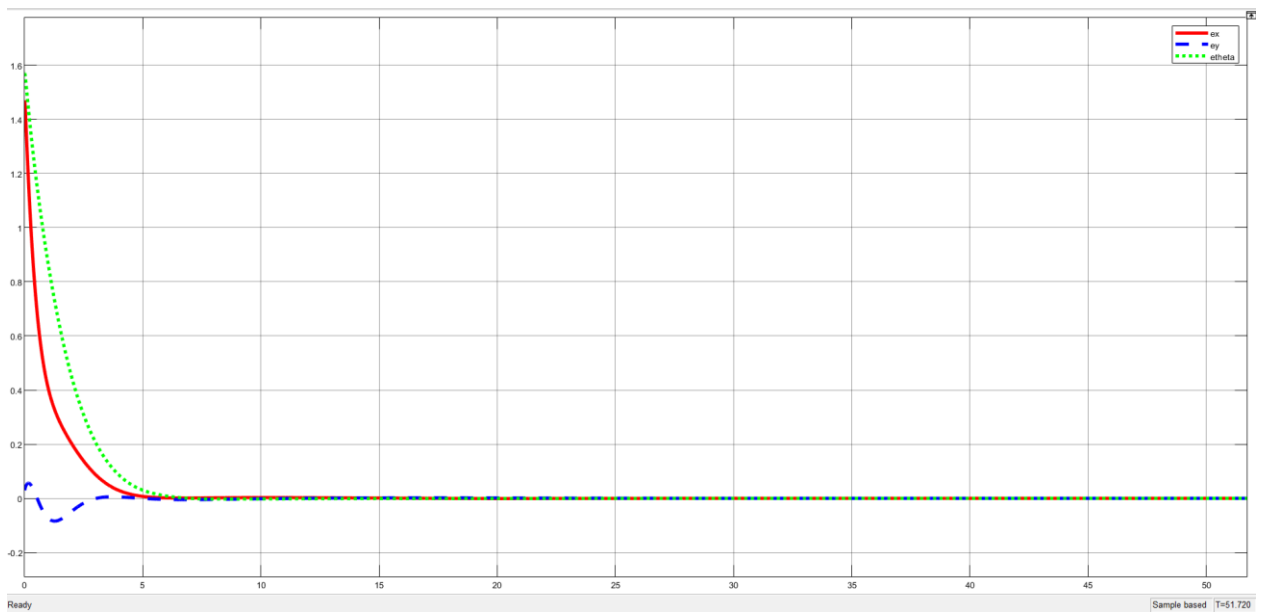
$$K_p = 0.02, K_i = 0.00015, K_d = 0.01$$

CHƯƠNG 3. MÔ PHỎNG VÀ THỰC NGHIỆM

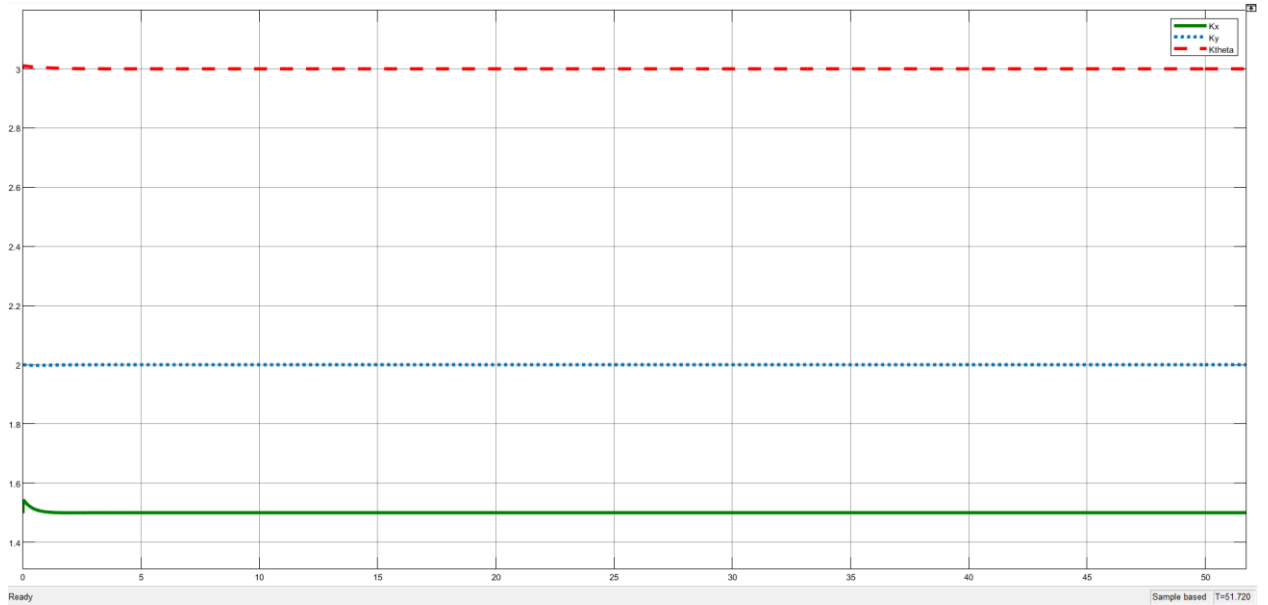
3.1 Kết quả mô phỏng



Hình 3.1 Quỹ đạo mô phỏng



Hình 3.2 Sai số mô phỏng



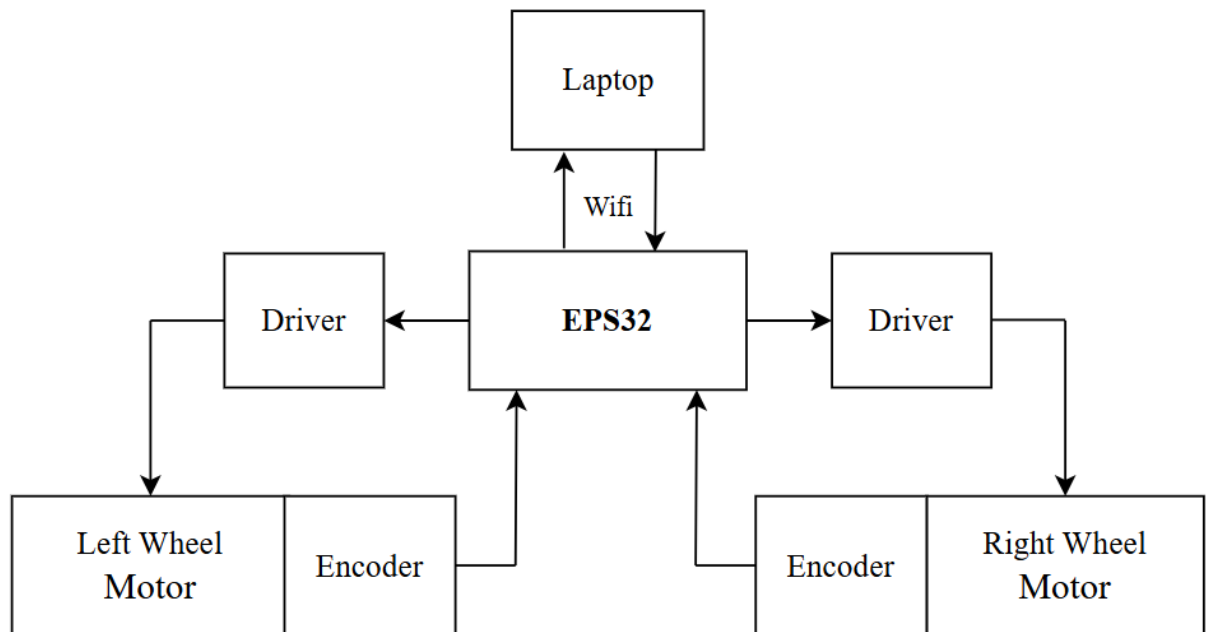
Hình 3.3 Các tham số được tinh chỉnh

3.2 Kết quả thực nghiệm

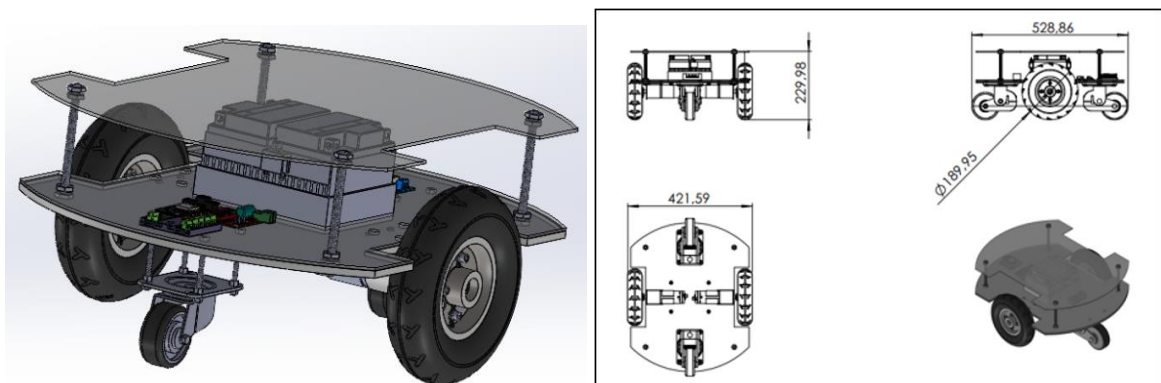
Trong phần này, chúng tôi thực hiện kiểm chứng bộ điều khiển đề xuất trong môi trường thực nghiệm. Vị trí ban đầu của robot được đặt tại gốc tọa độ (0, 0, 0); Quỹ đạo đặt là hình tròn được biểu diễn bởi phương trình (23):

$$\begin{cases} x_d = R \sin(\omega t) \\ y_d = R \cos(\omega t) \end{cases}; R = 1.25m, \omega = 0.23 \text{ rad/s} \quad (42)$$

Tham số của bộ điều khiển PID cho bộ phận truyền động được chọn dựa trên phương pháp Ziegler–Nichols thứ 2: $K_p = 0.002$, $K_i = 0.00015$, $K_d = 0.01$. Khoảng cách giữa 2 bánh xe là $2D=0.38m$, bán kính bánh xe: $r=0.1m$. Bộ điều khiển trung tâm được sử dụng là ESP32, động cơ DC servo được sử dụng ở đây là Planet 24VDC-60W với 13 xung encoder. Mạch XD-160D là mạch driver để điều khiển động cơ DC.



Hình 3.4 Sơ đồ khối



Hình 3.5 Bản vẽ mô hình



Hình 3.6 Mô hình thực nghiệm

Bộ thông số của mạng nơ ron:

Trọng số từ lớp đầu vào đến lớp ẩn.

Whi = [-6.4855 -4.0100 -4.5498;

-6.5890 -4.7984 -2.1504;

-6.5339 -3.9266 -4.3257;

-7.4260 -3.7515 -2.3720;

-6.3217 -4.2807 -3.3511;

-6.1753 -4.2071 -3.1082;

-8.9702 -5.8552 -4.7596;

-7.3099 -6.6322 -4.5009;

-6.2155 -5.9031 -5.4120;

-5.9862 -3.5501 -3.1355;

-5.9129 -4.1946 -3.8804;

-6.3110 -4.0644 -2.8849;

-6.2365 -2.9500 -2.3163;

-8.2328 -6.5682 -3.4148;

-7.1986 -3.6066 -2.8832;

-5.9831 -3.4436 -3.6779;

-6.3094 -3.8544 -3.2696;

-6.2915 -5.4654 -4.1758;

-6.5584 -4.4558 -3.2127;

-5.8346 -3.3768 -2.5838];

Trọng số từ lớp ẩn đến lớp đầu ra.

Woh= [0.0188 -1.9734 0.6255 0.4932 -0.5925 -0.4435 1.0799 0.3705 -0.1926 -1.1629

-0.0350 -0.9390 0.3647 -0.6056 0.5726 -0.1639 -1.3403 -1.5193 -1.6494 -1.8132

-0.2483;

-0.0693 1.2123 1.3129 1.6006 0.7402 1.1716 1.5867 3.1500 2.3577 3.3629 2.4780

2.6302 2.9674 0.7661 1.5265 2.0676 1.7297 4.0159 2.3748 2.4173 2.9054;

6.3216 3.3388 1.4826 -0.2754 -2.1694 -0.7430 -2.2963 -4.7001 -4.6418 -4.0895 -4.7556

-4.6468 -4.9112 -5.4908 -3.4376 -4.7539 -4.9947 -4.5363 -5.4131 -3.7736 -4.9652].

Số lượng đầu vào mạng nơ ron: Ni=2.

Số lượng đầu ra mạng nơ ron: No=3.

Số nơ ron trong lớp ẩn: Nh=20.

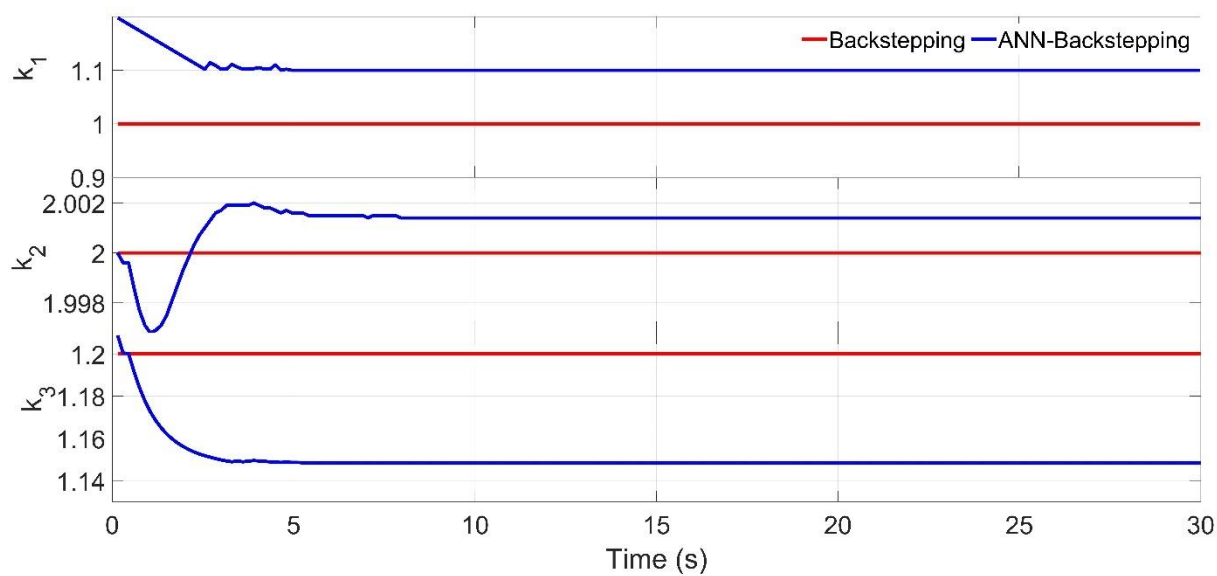
Hệ số tốc độ học: 0.7

Hệ số động lượng: 0.9

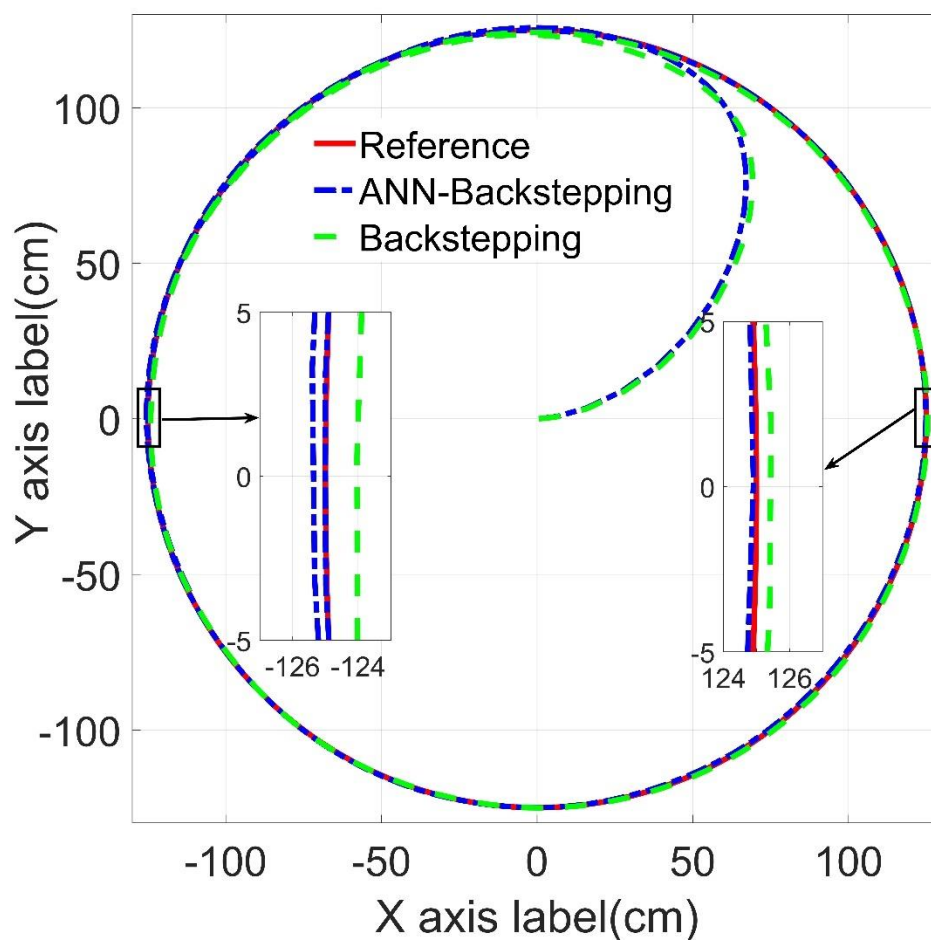
Kết quả thực nghiệm:

Video quá trình theo dõi và bám quỹ đạo của robot được thể hiện trong link đính kèm:

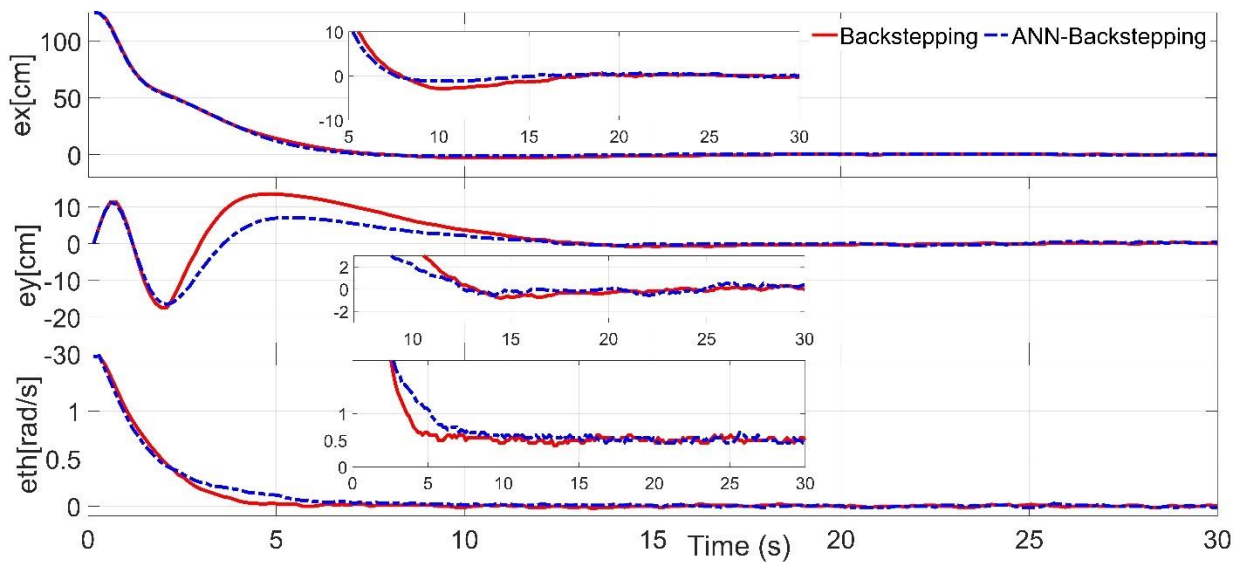
https://drive.google.com/drive/folders/1BHqT3GcOgGo_uNxxXz5bmAFDBDSqFcw2



Hình 3.7 Tham số điều khiển được huấn luyện và chỉnh định theo điều kiện thực nghiệm



Hình 3.8 Quỹ đạo thực nghiệm



Hình 3.9 Sai lệch đo được trong quá trình theo dõi quỹ đạo đặt

Các tham số điều khiển thay đổi liên tục trong quá trình thực nghiệm (Hình 3.7), đảm bảo rằng robot có thể phản ứng nhanh chóng với các biến động từ môi trường và duy trì sự ổn định trong quá trình bám quỹ đạo. Điều này khẳng định khả năng tự học và thích nghi của bộ điều khiển dựa trên mạng nơ-ron.

Sai lệch bám rất nhỏ và không có sự dao động lớn trên Hình 3.9, chứng minh rằng mạng nơ-ron đã giúp điều chỉnh và bù trừ sai lệch một cách hiệu quả, giúp duy trì sự ổn định của robot trong suốt quá trình vận hành. Từ Hình 3.8, có thể thấy rằng sai số bám quỹ đạo của robot trung bình là 2.5%, với sai số tối đa trong các thử nghiệm là 3.2%. Nguyên nhân của các sai số này chủ yếu đến từ nhiễu ngẫu nhiên trong môi trường thực nghiệm và sự thay đổi bất thường trong điều kiện mặt đất. Tuy nhiên, nhờ khả năng tự học của mạng nơ-ron, bộ điều khiển đã nhanh chóng điều chỉnh tham số và giảm thiểu sai số xuống mức thấp nhất. Kết quả này cho thấy robot có thể thích ứng và duy trì độ chính xác cao trong điều kiện môi trường thực nghiệm.

Kết quả thực nghiệm cũng cho thấy bộ điều khiển mạng nơ-ron thích nghi đạt hiệu suất cao hơn so với Backstepping trong điều kiện thực tế:

Bộ điều khiển mạng nơ-ron có sai số bám quỹ đạo trung bình thấp hơn so với Backstepping, với mức sai số trung bình là 2.5% so với 3.8% của Backstepping.

Thời gian hội tụ của bộ điều khiển mạng nơ-ron cũng ngắn hơn, trung bình 5 giây, trong khi Backstepping cần 7 giây để đạt độ ổn định. Điều này cho thấy khả năng phản ứng nhanh hơn của mạng nơ-ron đối với các thay đổi trong điều kiện môi trường.

CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

Trong đồ án này, nhóm đã phát triển và kiểm chứng một bộ điều khiển thích nghi dựa trên mạng nơ-ron cho WMR trong môi trường thực nghiệm. Bộ điều khiển này có khả năng tự học và thích nghi với các thay đổi của môi trường, giúp cải thiện đáng kể hiệu suất bám quỹ đạo và giảm sự phụ thuộc vào các mô hình động lực học phức tạp. Kết quả thực nghiệm đã cho thấy bộ điều khiển không chỉ đảm bảo độ chính xác cao trong quá trình bám quỹ đạo mà còn giúp robot hoạt động ổn định trong môi trường thực tế ngay cả khi hiểu biết về động lực học là hạn chế.

Việc sử dụng mạng nơ-ron để dự đoán và điều chỉnh các tham số điều khiển trong thời gian thực đã chứng minh hiệu quả vượt trội, với sai số trong quá trình bám quỹ đạo được giảm thiểu tối đa. Các kết quả từ *Hình 3.7, 3.8, 3.9* cho thấy robot đã bám sát quỹ đạo với sai số rất nhỏ, đồng thời bộ điều khiển đã tự điều chỉnh linh hoạt để đảm bảo tính ổn định trong toàn bộ quá trình vận hành. Đồ án này là tiền đề cho nhóm phát triển các hệ thống điều khiển thông minh trong thời gian tới, khi khả năng tự học và thích nghi là yếu tố then chốt để đối phó với các thách thức từ môi trường thực tế.

4.2 Hướng phát triển

Để tiếp tục nâng cao hiệu suất và khả năng ứng dụng của robot di động, các hướng phát triển trong tương lai có thể bao gồm:

- Thử Nghiệm Các Bộ Điều Khiển Mới: Để đánh giá hiệu quả và ổn định của hệ thống, có thể thêm các bộ điều khiển tiên tiến như Model Predictive Control (MPC) và các bộ điều khiển phi tuyến. Những bộ điều khiển này có thể giúp tối ưu hóa quá trình điều khiển, cải thiện độ chính xác và khả năng dự đoán của robot di động trong các điều kiện môi trường phức tạp.
- Tích Hợp Công Nghệ Xử Lý Ảnh: Nâng cao độ chính xác định vị bằng cách tích hợp công nghệ xử lý ảnh hoặc sử dụng thuật toán Kalman Filter kết hợp GPS và IMU. Điều này sẽ giúp robot xác định vị trí một cách chính xác hơn, đặc biệt trong các môi trường có nhiều nhiễu loạn hoặc thay đổi liên tục.
- Nâng Cấp Phần Cứng: Để xử lý dữ liệu phức tạp và yêu cầu tính toán cao, có thể nâng cấp phần cứng sử dụng trong hệ thống điều khiển như STM32 hoặc

Raspberry Pi. Những nâng cấp này sẽ cải thiện khả năng xử lý của hệ thống, đảm bảo robot có thể hoạt động mượt mà và hiệu quả hơn trong các ứng dụng thực tế.

Những hướng phát triển này không chỉ giúp nâng cao chất lượng và hiệu quả của robot di động mà còn mở ra những cơ hội ứng dụng mới trong nhiều lĩnh vực khác nhau, từ công nghiệp đến dịch vụ và nghiên cứu khoa học.

TÀI LIỆU THAM KHẢO

- [1] F. N. Martins, W. C. Celeste, R. Carelli, M. Sarcinelli-Filho, "A neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Transactions on Neural Networks*, vol. 19, no. 5, pp. 1178–1188, 2008.
- [2] S. Li, Y. Wang, Y. Luo, "Robust adaptive control for wheeled mobile robots using neural networks," *Journal of Intelligent & Robotic Systems*, vol. 50, pp. 169–183, 2007.
- [3] L. Li, T. Wang, Y. Xia, and N. Zhou, "Trajectory tracking control for wheeled mobile robots based on nonlinear disturbance observer with extended Kalman filter," *Journal of the Franklin Institute*, vol. 357, no. 15, pp. 8491–8507, 2020.
- [4] Štefek, V. T. Pham, V. Krivanek, and K. L. Pham, "Optimization of Fuzzy Logic Controller Used for a Differential Drive Wheeled Mobile Robot," *Applied Sciences*, vol. 11, no. 13, pp. 6023, 2021.
- [5] P. Panahandeh, K. Alipour, B. Tarvirdizadeh, A. Hadi, "A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots," *Mechanical Systems and Signal Processing*, vol. 134, p. 106319, 2019.
- [6] Y. Guo, "Fuzzy control and neural network-based trajectory tracking for non-holonomic mobile robots," *Journal of Control Engineering and Technology*, vol. 12, no. 2, pp. 189–196, 2019.
- [7] M. Abdollahi, M. M. Baygi, M. A. Ahmadi, "Trajectory tracking control for mobile robots with neural network disturbance estimation," *International Journal of Robotics and Automation*, vol. 33, pp. 129–136, 2015.
- [8] Rossi, M. Milani, "Trajectory tracking via robust adaptive control of a non-holonomic mobile robot," *Journal of Dynamic Systems, Measurement, and Control*, vol. 140, 2018(2020-(J)_Controlling wh...).
- [9] Q. V. Dang, I. Nielsen, S. Bøgh, G. Bocewicz, "Modelling and scheduling autonomous mobile robot for a real-world industrial application," *IFAC Proceedings Volumes*, vol. 46(9), pp. 2098–2103, 2013.
- [10] Pazderski, "Application of transverse functions to control differentially driven wheeled robots using velocity fields," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 64.4, 2016.
- [11] L. Armesto, V. Girbés, A. Sala, M. Vicente, "Adaptive tracking for wheeled mobile robots using fuzzy logic," *International Journal of Approximate Reasoning*, vol. 45, pp. 163–179, 2018.
- [12] H. S. Phuong, N. M. Tien, N. D. Tan, M. T. Anh and D. D. Tu, "Proposal of a Fault-tolerant controller for wheeled mobile robots with faulty actuators", 12th International Conference on Control, Automation and Information Sciences (ICCAIS), pp. 507-512, 2023.