

# Kiểm chứng thực nghiệm bộ điều khiển thích nghi mạng nơ-ron cho robot di động bám quỹ đạo

Hồ Sỹ Phương<sup>1,2\*</sup>, Dương Đình Tú<sup>1</sup>, Trần Phương Nam<sup>1</sup>, Nguyễn Thành Trí<sup>1</sup>, Lê Thị Thu Uyên<sup>1</sup>, Đặng Thái Sơn<sup>1</sup>, Phan Văn Vỹ<sup>1</sup>

<sup>1</sup>Viện Kỹ thuật và Công nghệ, Trường Đại học Vinh, <sup>2</sup>Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Email: hophuong@vinhuni.edu.vn, tranphuongnam292003@gmail.com

**Abstract**— Trong bài báo này, chúng tôi đề xuất một mô hình điều khiển thích nghi cho robot di động hai bánh vi sai (WMR) dựa trên mạng nơ-ron để bám quỹ đạo. Đóng góp chính của nghiên cứu nằm ở việc kiểm chứng bộ điều khiển trực tiếp trên mô hình robot thực tế thay vì chỉ dựa vào mô phỏng. Bộ điều khiển sử dụng mạng nơ-ron có khả năng tự động học hỏi từ dữ liệu thời gian thực và điều chỉnh theo sự thay đổi của môi trường, nhờ đó cải thiện hiệu suất điều khiển và giảm sự phụ thuộc vào các mô hình động lực học phức tạp. Kết quả cho thấy bộ điều khiển dựa trên mạng nơ-ron đạt độ chính xác cao trong việc bám quỹ đạo trong môi trường thực nghiệm với hiểu biết về động lực học là hạn chế. Bài báo cũng trình bày chi tiết quá trình thiết kế, triển khai và các kết quả thực nghiệm để chứng minh tính hiệu quả và độ tin cậy của phương pháp.

**Keywords**- Robot di động có bánh xe, điều khiển thích nghi, mạng nơ-ron, bám quỹ đạo, học thời gian thực.

## I. GIỚI THIỆU

Robot di động 2 bánh vi sai (DDMR) được sử dụng rất phổ biến trong các ứng dụng công nghiệp và dịch vụ. Tuy nhiên, việc điều khiển chính xác quỹ đạo của robot trong môi trường thực tế gặp nhiều thách thức 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 hoặc điều khiển trượt (SMC) được áp dụng rộng rãi, nhưng chúng phụ thuộc nhiều vào các mô hình động lực học chính xác của hệ thống, làm cho việc đối phó với các nhiễu động và biến đổi không dự đoán trước trở nên khó khăn [3][4].

Để khắc phục những hạn chế này, mạng nơ-ron nhân tạo (ANN) đã được sử dụng trong các bộ điều khiển để học và thích nghi với môi trường thay đổi, cải thiện khả năng bám quỹ đạo của robot mà không cần dựa vào các mô hình động lực học phức tạp [5]. Các nghiên cứu trước đây đã cho thấy mạng nơ-ron có khả năng ước lượng các tham số hệ thống và xử lý tốt các yếu tố bất định trong môi trường hoạt động của robot [6][7]. Tuy nhiên, nhiều nghiên cứu chỉ tập trung vào mô phỏng mà chưa thử nghiệm trực tiếp trên các hệ thống thực tế [8].

Trong nghiên cứu này, chúng tôi thực hiện kiểm chứng một bộ điều khiển thích nghi dựa trên mạng nơ-ron cho robot di động. Bộ điều khiển này không chỉ

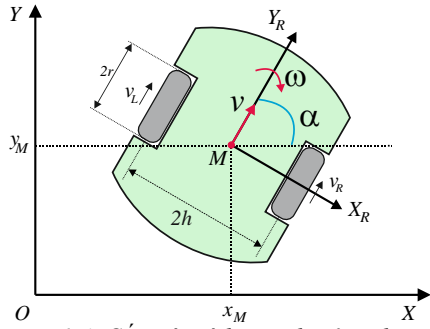
bám quỹ đạo chính xác trong môi trường có nhiễu và các yếu tố biến đổi mà còn cải thiện độ ổn định và giảm thiểu sai số [9]. Bộ điều khiển được thiết kế với khả năng tự học và điều chỉnh các tham số động lực học chưa biết thông qua mạng nơ-ron, giúp tối ưu hóa quá trình điều khiển và giảm phụ thuộc vào các mô hình toán học [10].

Kết quả thử nghiệm cho thấy bộ điều khiển thích nghi này đạt được hiệu suất cao trong bám quỹ đạo và đảm bảo tính ổn định của hệ thống ngay cả trong môi trường thực tế có nhiễu động lớn [11]. Nghiên cứu này không chỉ cung cấp một giải pháp điều khiển hiệu quả cho robot di động mà còn mở ra tiềm năng ứng dụng trong các hệ thống điều khiển tự động khác.

Nội dung nghiên cứu gồm 5 phần, phần 2 giới thiệu về mô hình hóa robot và bản thiết kế phần cứng của robot. Phần 3 trình bày bộ điều khiển đề xuất với khả năng tự chỉnh định các tham số điều khiển nhờ sử dụng mạng nơ-ron. Phần 4 trình bày các kết quả đạt được khi áp dụng thuật toán trên mô hình thực nghiệm, cuối cùng là kết luận thể hiện trong phần 5.

## II. MÔ HÌNH ROBOT

Mô hình được nghiên cứu, thiết kế và chế tạo ở đây có cấu trúc cơ bản thể hiện trên hình 1. Robot chuyển động phụ thuộc hoàn toàn vào bộ truyền động của 2 bánh xe bán kính  $r$  gắn trên 2 động cơ DC servo được bố trí đồng trục. Vận tốc góc và vận tốc dài tương ứng của bánh phải và bánh trái được ký hiệu lần lượt là  $\omega_i, v_i$  ( $i = R, L$ ).  $\zeta = [v \ \omega]^T$  là véc tơ thể hiện vận tốc dài của robot. bánh thụ động phía trước sau giữ vai trò thăng bằng cho robot. Bề rộng của robot là  $2h$ , trọng tâm và tâm của robot được thiết kế trùng nhau tại M.  $q = [x_M \ y_M \ \alpha_M]^T$ ;  $q_d = [x_d \ y_d \ \alpha_d]^T$  là vị trí tương ứng của robot trong 2 hệ tọa độ (hệ tọa độ toàn cục và di động được thể hiện tương ứng trên hình 1).



Hình 1. Cấu trúc và hệ tọa độ của robot

Vị trí tương ứng của robot trong hai hệ tọa độ bao gồm:  $q = [x_M \ y_M \ \alpha_M]^T$ ;  $q_d = [x_d \ y_d \ \alpha_d]^T$

Mối quan hệ giữa vận tốc của robot trong hai hệ tọa độ được thể hiện thông qua phương trình:

$$\dot{q} = S \dot{q}_d \Leftrightarrow \begin{bmatrix} \dot{x}_M \\ \dot{y}_M \\ \dot{\alpha}_M \end{bmatrix} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\alpha}_d \end{bmatrix} \quad (1)$$

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.

Vận tốc dài của mỗi bánh xe:

$$\begin{cases} v_R = r\omega_R \\ v_L = r\omega_L \end{cases} \quad (2)$$

Thành phần vận tốc của robot:

$$\begin{cases} v = \frac{v_R + v_L}{2} = \frac{r}{2}(\omega_R + \omega_L) \\ \omega = \frac{v_R - v_L}{2h} = \frac{r}{2h}(\omega_R - \omega_L) \end{cases} \quad (3)$$

Với giả thiết robot di động không có trượt ngang và trượt dọc, 2 bánh vi sai luôn tiếp xúc với mặt đất. Vận tốc tại điểm “M” sẽ được tính như sau:

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\alpha}_d \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1/h & -1/h \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (4)$$

Kết hợp (1), (3) và (4), phương trình động học của robot được biểu diễn như sau:

$$\dot{q} = H \zeta \quad (5)$$

Trong đó:

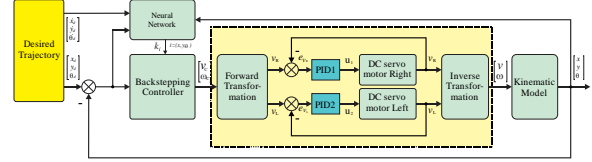
$$H = \begin{bmatrix} \cos \alpha & 0 \\ \sin \alpha & 0 \\ 0 & 1 \end{bmatrix}; \zeta = \begin{bmatrix} v \\ \omega \end{bmatrix}$$

### III. THIẾT KẾ BỘ ĐIỀU KHIỂN THÍCH NGHI TRÊN CƠ SỞ MẠNG NƠ-RON

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 hồi 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 với sự tham gia của bộ điều khiển động học với khả năng thích nghi nhờ sử dụng kiến trúc mạng nơ-ron thể hiện trong hình 3.



Hình 2. Sơ đồ cấu trúc bộ điều khiển đề xuất

Phương trình sai lệch bám vị trí của robot được chỉ ra trong phương trình (6) [13]:

$$e_q = \begin{bmatrix} \xi_x \\ \xi_y \\ \xi_\alpha \end{bmatrix} = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x_M \\ y_d - y_M \\ \alpha_d - \alpha_M \end{bmatrix} \quad (6)$$

Đạo hàm của 2 vế ta có:

$$\begin{cases} \dot{\xi}_x = -v + \omega \xi_y + v_d \cos \xi_\alpha \\ \dot{\xi}_y = -\omega \xi_x + v_d \sin \xi_\alpha \\ \dot{\xi}_\alpha = \omega_d - \omega \end{cases} \quad (7)$$

Luật điều khiển động học được đưa ra và áp dụng cho robot được thể hiện qua (8) [12].

$$\zeta_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_d \cos \xi_\alpha + \lambda_1 \xi_x \\ \omega_d + v_d (\lambda_2 \xi_y + \lambda_3 \sin \xi_\alpha) \end{bmatrix} \quad (8)$$

Trong đó  $\lambda_i (i=1,2,3)$  là các giá trị dương. Trong nghiên cứu này chúng được cập nhật một cách liên tục bằng mạng nơ-ron giúp robot có khả năng thích ứng tốt hơn trước các nhiễu loạn và các nhân tố ảnh hưởng chưa biết.

Sự điều chỉnh  $\lambda_i$  của bộ điều khiển dựa trên động học để tối thiểu hóa hàm chi phí sau:

$$L = \frac{1}{2} \xi_q^T \gamma \xi_q \quad (9)$$

Trong đó:  $\gamma = \text{diag}(\gamma_1, \gamma_2, \gamma_3)$  là các hệ số của mạng nơ-ron;  $\lambda_i$  là một phần của hàm chi phí và được tối ưu hóa cũng như cập nhật theo phương pháp gradient descent. Các hệ số này được biểu diễn bởi vector  $\kappa = [\lambda_1 \ \lambda_2 \ \lambda_3]^T$ ,

Kết hợp với (6), đạo hàm riêng của hàm chi phí  $\Gamma$  theo  $\kappa$  được tính như sau:

$$\frac{\partial \Gamma}{\partial \kappa} = \xi_q^T \gamma \frac{\partial \xi_q}{\partial \kappa} = -\xi_q^T \gamma T \frac{\partial q}{\partial \kappa} \quad (10)$$

Trong đó:

$$T = \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

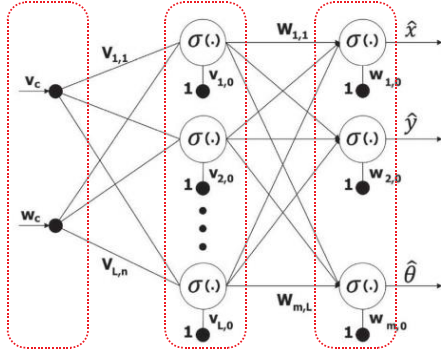
Áp dụng quy tắc chuỗi, đạo hàm của  $q$  theo  $\kappa$  được khai triển:

$$\frac{\partial q}{\partial \kappa} = \frac{\partial q}{\partial \zeta_c} \frac{\partial \zeta_c}{\partial \kappa} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \\ J_{31} & J_{32} \end{bmatrix} \begin{bmatrix} \xi_x & 0 & 0 \\ 0 & v_d \xi_y & v_d \sin \xi_\alpha \end{bmatrix} \quad (11)$$

Lúc đó, (10) được viết lại như sau:

$$\frac{\partial \Gamma}{\partial \kappa} = \xi_q^T \gamma \frac{\partial \xi_q^T}{\partial \kappa} = -\xi_q^T \gamma T J \frac{\partial \zeta_c}{\partial \kappa} \quad (12)$$

Trong đó:  $J = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \\ J_{31} & J_{32} \end{bmatrix}$



Hình 3. Cấu trúc các lớp của mạng nơ-ron

Phương pháp tính ma trận Jacobian bằng mạng nơ-ron (NN) được thực hiện bằng cách khởi tạo các trọng số ban đầu cho mô hình, sau đó tinh chỉnh các trọng số này thông qua quá trình huấn luyện. Mạng nơ-ron dự đoán các giá trị thông qua lan truyền xuôi, từ đó xác định sai số bằng cách so sánh giá trị dự đoán với giá trị thực. Sai số này được lan truyền ngược từ lớp đầu ra về lớp đầu vào, sử dụng các đạo hàm để điều chỉnh trọng số, tối ưu hóa mô hình và từ đó tính toán chính xác ma trận Jacobian. Quy trình này đảm bảo mạng nơ-ron có thể học và thích nghi hiệu quả trong các hệ thống điều khiển.

Lớp đầu tiên của mô hình mạng nơ-ron nhận  $L$  nơ-ron, lớp thứ 2 là  $m$  nơ-ron.  $\zeta_c$  và  $\hat{x}, \hat{y}, \hat{\alpha}$  tương ứng là đầu vào và đầu ra cho mạng nơ-ron.  $N_h$  là số nơ-ron trong lớp ẩn. Trong khi các giá trị  $Wh_i$  là trọng số từ lớp ẩn tới lớp đầu vào, thì các giá trị  $Who$  là trọng số từ lớp đầu ra đến lớp ẩn.

Mối quan hệ giữa các đầu vào ( $x_j$ ) và đầu ra ( $y_i$ ) của mạng được xác định bởi:

$$y_i = \sigma \left( \sum_{j=1}^L w_{oh_i} \sigma \left( \sum_{j=1}^2 w_{hi_j} x_j + v_{i0} \right) + w_{i0} \right) \quad (13)$$

Hàm kích hoạt được chọn là hàm sigmoid:

$$\sigma = \frac{1}{1 + e^{-\rho}} \quad (14)$$

Trong đó  $\rho$  là biểu diễn của tích giữa trọng số và giá trị neuron dẫn tới lớp mong muốn. Đạo hàm của hàm 2 về của (14) ta có:

$$\frac{\partial \sigma}{\partial \rho} = \sigma(1 - f(\rho)) \quad (15)$$

Thuật toán lan truyền ngược là một thuật toán điều chỉnh trọng số dựa trên phương pháp giảm dần gradient.

$$\begin{cases} W_{il}(k+1) = W_{il}(k) - \varsigma \frac{\partial E(k)}{\partial W_{il}} \\ V_{ij}(k+1) = V_{ij}(k) - \varsigma \frac{\partial E(k)}{\partial V_{ij}} \end{cases} \quad (16)$$

Trong đó  $\varsigma$  là hệ số học (learning rate), nó quyết định bước nhảy trong quá trình giảm thiểu sai lệch (gradient descent). Trọng số được thay đổi theo chu kỳ cho đến khi sai lệch được tối thiểu hóa.

Sai lệch của robot cần tối thiểu hóa được xác định:

$$e_i(k) = Y_i(k) - y_i(k) \quad (17)$$

Trong đó  $y_i(k)$  là giá trị đầu ra của mạng nơ-ron tại bước  $k$ ,  $Y_i(k) = [x_A \ y_A \ \alpha_A]^T$  là vị trí và góc lái của robot tại một thời điểm A cụ thể, đóng vai trò làm giá trị tham chiếu để so sánh với các giá trị dự đoán của hệ thống.

Bằng cách sử dụng hàm lỗi, hàm chi phí được tìm thấy như sau:

$$E(k) = \frac{\xi_i(k)^2}{2} \quad (18)$$

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).

$$\begin{cases} \frac{\partial E}{\partial W_{il}} = -z_l \sigma'(\rho_i^2) \xi_i \\ \frac{\partial E}{\partial V_{ij}} = -x_j \sigma'(\rho_i^1) \sum (\sigma'(\rho_i^2) \xi_i + W_{il}) \end{cases} \quad (19)$$

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

$$J = \begin{bmatrix} \frac{\hat{x}(t) - \hat{x}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{x}(t) - \hat{x}(t-1)}{w_c(t) - w_c(t-1)} \\ \frac{\hat{y}(t) - \hat{y}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{y}(t) - \hat{y}(t-1)}{w_c(t) - w_c(t-1)} \\ \frac{\hat{\alpha}(t) - \hat{\alpha}(t-1)}{v_c(t) - v_c(t-1)} & \frac{\hat{\alpha}(t) - \hat{\alpha}(t-1)}{w_c(t) - w_c(t-1)} \end{bmatrix} \quad (20)$$

Thông qua mạng nơ-ron, bộ điều khiển động học (8) thay đổi các giá trị điều chỉnh và thích nghi để làm cho hàm chi phí bằng 0 theo phương pháp giảm dần gradient như sau:

$$\begin{cases} k_1(t) = k_1(t-1) + \Delta k_1 \\ k_2(t) = k_2(t-1) + \Delta k_2 \\ k_3(t) = k_3(t-1) + \Delta k_3 \end{cases} \quad (21)$$

Trên cơ sở tín hiệu điều khiển động học với khả năng tự điều chỉnh, bộ điều khiển PID được sử dụng để điều khiển hệ truyền động của robot trên cơ sở so sánh sai số vận tốc của robot thực tế với giá trị điều khiển. Cấu trúc bộ điều khiển PID sử dụng có dạng:

$$u_m = K_p \xi_\eta + K_I \int \xi_\eta dt + K_D \dot{\xi}_\eta \quad (22)$$

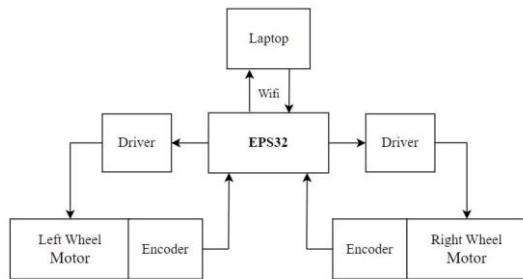
Trong đó  $K_p, K_I, K_D$  là các hệ số điều khiển dương tương ứng với các khâu trong bộ điều khiển PID.

#### IV. KẾT QUẢ VÀ THẢO LUẬN

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:

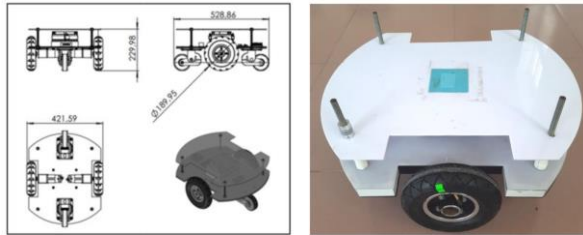
$$\begin{cases} x_d = R \sin(\omega t) \\ y_d = R \cos(\omega t) \end{cases}; R = 1.25m, \omega = 0.25 \text{ rad/s} \quad (23)$$

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



Hình 4. Sơ đồ cấu trúc ghép nối phần cứng

Mô hình thực nghiệm của robot được thể hiện trên hình 5.



Hình 5. Thiết kế và mô hình thực tế của robot

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.

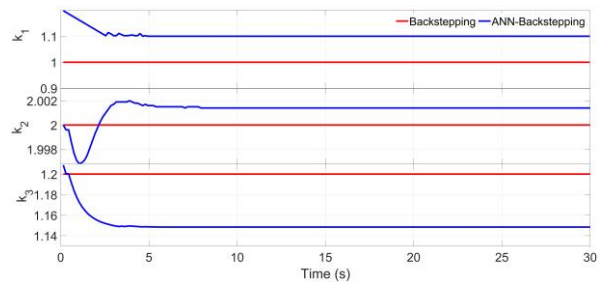
Số nơ ron trong lớp ẩn: Nh=20.

Hệ số tốc độ học: 0.7

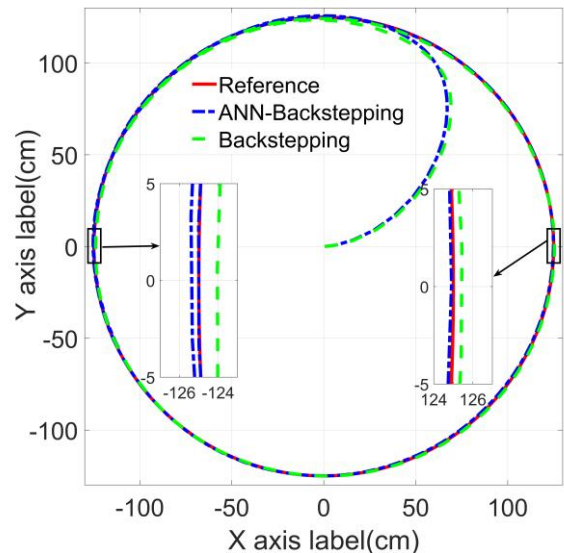
**Kết quả thực nghiệm:**

Quá trình chạy thực nghiệm, nhóm tác giả có sử dụng thêm một bộ điều khiển Backstepping để tham chiếu và lấy kết quả so sánh. Video quá trình theo dõi và bám quỹ đạo của robot được thể hiện trong link đính kèm:

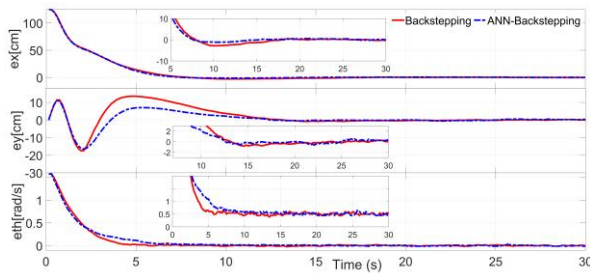
[https://drive.google.com/drive/folders/1BHqT3GcOgGo\\_uNxxXz5bmAFDBDSqFw2](https://drive.google.com/drive/folders/1BHqT3GcOgGo_uNxxXz5bmAFDBDSqFw2)



Hình 6. Tham số điều khiển được huấn luyện và chỉnh định theo điều kiện thực nghiệm



Hình 7. Quỹ đạo thực nghiệm.



Hình 8. 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 6), 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 8, 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 7, 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.

## V. KẾT LUẬN

Trong nghiên cứu này, chúng tôi 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 5, 6, và 7 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. Nghiên cứu này là tiền đề cho nhóm nghiên cứu 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.

## VI. 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.