

Performance Evaluation of a PID Sliding Surface and Radial Basis Function Neural Network for Mobile Robot

Trung Hieu Tran¹, Viet Trung Nguyen¹, Thuy Trang Tran Thi¹, Thanh Tung Pham^{*1}

Vinh Long University of Technology Education, Vietnam

*Corresponding author. Email: tungpt@vlute.edu.vn

ARTICLE INFO

Received: 15/12/2023
Revised: 04/01/2024
Accepted: 27/03/2024
Published: 28/10/2024

KEYWORDS

PID;
Radial basis function neural network;
Sliding mode control;
Mobile robot;
MATLAB/Simulink.

ABSTRACT

A proportional integral derivative sliding surface (PIDSS) and radial basis function neural network (RBF-NN) for Mobile robot are applied in this study. This robot has many advantages such as simple structure, energy saving, high moving speed, and low production costs. The sliding mode control (SMC) controller using PIDSS (PIDSS-SMC) is designed so that the robot's actual output approaches the standard input and reduces the chattering phenomena around the sliding surface. The RBF-NN is used to approximate the nonlinear components in the P_w matrix of the PIDSS-SMC controller. The weights of this neural network are trained online using the Gradient Descent algorithm. Lyapunov theory is used to prove the stability of the system. The actual output of the x_w and y_w converges to the reference x_d and y_d with the steady-state error converges to zero, the rising time reaches 0.0832s and 0.0764s, the settling time is 0.1309s and 0.1226s, the overshoot is 0.0042% and 0.0055%, respectively, and the chattering phenomena was reduced.

Đánh Giá Hiệu Quả Mặt Trượt PID và Mạng RBF-NN Cho Mobile Robot

Trần Trung Hiếu¹, Nguyễn Việt Trung¹, Trần Thị Thùy Trang¹, Phạm Thanh Tùng^{*1}

Đại học Sư phạm Kỹ thuật Vinh Long, Việt Nam

*Tác giả liên hệ. Email: tungpt@vlute.edu.vn

THÔNG TIN BÀI BÁO

Ngày nhận bài: 15/12/2023
Ngày hoàn thiện: 04/01/2024
Ngày chấp nhận đăng: 27/03/2024
Ngày đăng: 28/10/2024

TỪ KHÓA

PID;
Mạng nơ-ron hàm cơ sở xuyên tâm;
Điều khiển trượt;
Mobile robot;
MATLAB/Simulink.

TÓM TẮT

Mặt trượt vi tích phân tỷ lệ và mạng RBF-NN cho Mobile robot được ứng dụng trong nghiên cứu này. Đây là robot có nhiều ưu điểm như cấu trúc đơn giản, tiết kiệm năng lượng, tốc độ di chuyển cao, và chi phí sản xuất thấp. Bộ điều khiển trượt sử dụng mặt trượt PID (PIDSS-SMC) được thiết kế sao cho ngõ ra thực tế của robot tiến về ngõ vào chuẩn và giảm hiện tượng chattering quanh mặt trượt. Mạng RBF-NN được sử dụng để xấp xỉ các thành phần phi tuyến trong ma trận P_w của bộ điều khiển PIDSS-SMC. Các trọng số của mạng được huấn luyện trực tuyến sử dụng giải thuật Gradient Descent. Lý thuyết Lyapunov được sử dụng để chứng minh tính ổn định của hệ thống. Ngõ ra thực tế của x_w và y_w hội tụ về x_d và y_d tham chiếu với sai số xác lập hội tụ về 0, thời gian tăng đạt 0,0832s và 0,0764s; thời gian xác lập là 0,1309s và 0,1226s; độ vọt lố là 0,0042% và 0,0055% tương ứng, và hiện tượng chattering được giảm.

Doi: <https://doi.org/10.54644/jte.2024.1505>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Giới thiệu

Mobile robot đã được sử dụng trong phạm vi rộng của các lĩnh vực, trong không gian hẹp đòi hỏi tính di động cao trong các nhà máy và bệnh viện. Đặc điểm chính của robot là trực tiếp di chuyển tới một hướng tùy ý mà không thay đổi hướng [1]. Sự di động vốn có của robot này khiến nó được nghiên cứu rộng rãi để ứng dụng môi trường năng động [2]. Điều khiển bám quỹ đạo chính xác là thành phần

quan trọng trong các ứng dụng của Mobile robot, chẳng hạn như trong nghiên cứu [3] đã thiết kế bộ điều khiển robot sử dụng phương pháp điều khiển tuyến tính hóa quỹ đạo dựa vào mô hình động học của robot. Phương pháp PID mờ được phát triển trong [4] với thời gian tiến về quỹ đạo chuẩn khoảng 3(s). Trong [5] đã triển khai cách thức điều khiển mô-men xoắn để đối trọng với các điều kiện phi tuyến của robot. Nghiên cứu [6] đề xuất bộ điều khiển PI cho 2 vòng lặp để điều khiển robot di chuyển tịnh tiến theo quỹ đạo cho trước. Trong [7] đã thử nghiệm bộ điều khiển RBF-PD trên mô hình robot để bù thành phần bất định trên hệ thống phi tuyến. Trong nghiên cứu [8] đã so sánh hiệu quả của bộ điều khiển trượt và trượt – RBF cho mobile robot. Mục tiêu chính của [9] là điều khiển bền vững thích nghi cho robot với thành phần bất định như: độ lệch điều kiện ban đầu, khối lượng và mô men quán tính của hệ thống thay đổi theo thời gian và bị chặn. Mạng nơ-ron thích nghi được áp dụng trong sơ đồ điều khiển phi tuyến dựa trên Lyapunov để xử lý các lực ma sát và các động lực học không được mô hình hóa khác hoặc các nhiễu loạn bên ngoài có thể xảy ra [10], [11]. Bộ điều khiển trượt thích nghi được phát triển trong [12]. Các quy tắc thích nghi dựa trên RBFNN được giới thiệu cho gần đúng thành phần bất định của mô hình, nhiễu ngoài chưa biết. Thuật toán gradient chính sách xác định sâu (DDPG) [13] đã được thiết kế để huấn luyện trong môi trường có không gian hoạt động liên tục để bám theo quỹ đạo. Bộ điều khiển bám thích nghi cho robot [14] với các ràng buộc trạng thái đầy đủ, độ không đảm bảo của mô hình và nhiễu loạn bên ngoài. Bộ bù ước tính nhiễu bằng mạng nơ-ron hàm cơ sở xuyên tâm đã được đề xuất [15].

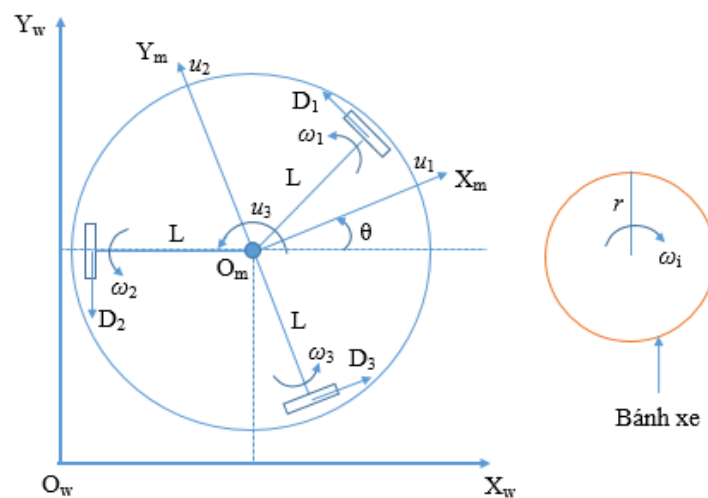
Mobile robot đã được nhiều nhà khoa học nghiên cứu với đa dạng các bộ điều khiển, trong đó bộ điều khiển trượt đã được đề cập. Điều đó cho thấy khả năng ứng dụng tích cực của điều khiển trượt với ưu điểm đáp ứng nhanh và bền vững với sự thay đổi của nhiễu cũng như các thông số của đối tượng. Tuy nhiên, khuyết điểm của kỹ thuật này chính là hiện chattering. Hiện tượng này sẽ phá hủy các cơ cấu chấp hành, làm cho các mạch điện tử quá nhiệt dẫn đến hư hỏng. Để giải quyết vấn đề này, nghiên cứu đề xuất sử dụng mặt trượt vi tích phân tỷ lệ (PIDSS) dựa vào mạng RBF-NN ứng dụng cho điều khiển bám quỹ đạo Mobile robot. Mặt trượt PID được thiết kế thay cho mặt trượt cổ điển góp phần giảm hiện tượng chattering quanh mặt trượt. Mạng RBF-NN được sử dụng để ước lượng các thành phần phi tuyến trong luật điều khiển PIDSS-SMC. Các trọng số của mạng được cập nhật trực tuyến căn cứ theo giải thuật Gradient Descent.

Bài báo này được tổ chức như sau: phần 2 trình bày phương pháp sử dụng mặt trượt PID và mạng RBF-NN cho Mobile robot, kết quả và bàn luận được trình bày trong phần 3 và phần 4 là kết luận.

2. Phương pháp sử dụng mặt trượt PID và RBF-NN cho Mobile robot

2.1. Phương trình toán học Mobile robot

Robot này có ba bánh xe như Hình 1 với mô hình toán học như (1) được sử dụng và kế thừa từ nghiên cứu [13], [16]:



Hình 1. Mobile robot [13], [16]

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} v_1 & -v_2\dot{\theta} & 0 \\ v_2\dot{\theta} & v_1 & 0 \\ 0 & 0 & v_3 \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \kappa_1\delta_1 & \kappa_1\delta_2 & 2\kappa_1\cos\theta \\ \kappa_1\delta_3 & \kappa_1\delta_4 & 2\kappa_1\sin\theta \\ \kappa_2 & \kappa_2 & \kappa_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} N_{fx} \\ N_{fy} \\ N_{f\theta} \end{bmatrix} \quad (1)$$

$$= \mathbf{P}_w \mathbf{X} + \mathbf{Q}_w \mathbf{U}_{con} + \mathbf{N}_f$$

với $\mathbf{N}_f = [N_{fx} \quad N_{fy} \quad N_{f\theta}]^T$ là nhiều chưa biết.

Trong đó:

$$\mathbf{P}_w = \begin{bmatrix} v_1 & -v_2\dot{\theta} & 0 \\ v_2\dot{\theta} & v_1 & 0 \\ 0 & 0 & v_3 \end{bmatrix}, \mathbf{Q}_w = \begin{bmatrix} \kappa_1\delta_1 & \kappa_1\delta_2 & 2\kappa_1\cos\theta \\ \kappa_1\delta_3 & \kappa_1\delta_4 & 2\kappa_1\sin\theta \\ \kappa_2 & \kappa_2 & \kappa_2 \end{bmatrix}, \mathbf{U}_{con} = [u_1 \quad u_2 \quad u_3]^T,$$

$$v_1 = \frac{-3c}{(3I_w + 2Mr^2)}; v_2 = \frac{2Mr^2}{(3I_w + 2Mr^2)}; v_3 = \frac{-3cL^2}{(3I_wL^2 + I_vr^2)}; \kappa_1 = \frac{kr}{(3I_w + 2Mr^2)}$$

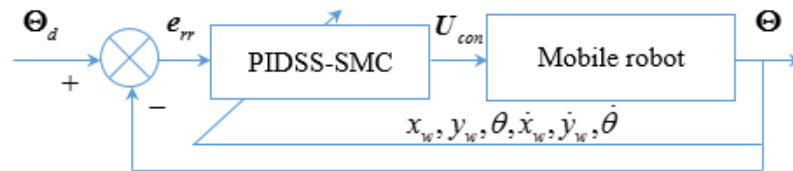
$$\kappa_2 = \frac{krL}{(3I_w + I_vr^2)}; v_2 = 1 - v_2 = \frac{3I_w}{(3I_w + 2Mr^2)}; \delta_1 = -\sqrt{3}\sin\theta - \cos\theta;$$

$$\delta_2 = \sqrt{3}\sin\theta - \cos\theta; \delta_3 = \sqrt{3}\cos\theta - \sin\theta; \delta_4 = -\sqrt{3}\cos\theta - \sin\theta$$

Trong đó, theo [13], [16], L là khoảng cách giữa bất kỳ bộ phận và tâm trọng lực của robot, c là hệ số ma sát nhớt cho bánh xe, D_i là lực lái cho mỗi bộ phận, r là bán kính của mỗi bánh xe, I_v là momen quán tính của mỗi bánh xe quanh trục lái, w_i là tốc độ quay của bánh xe, k là yếu tố độ lợi lái, u_i momen xoắn ngõ vào lái.

2.2. Ứng dụng mặt trượt PID cho Mobile robot

Nghiên cứu này áp dụng điều khiển trượt dựa vào mặt trượt vi tích phân tỷ lệ (PIDSS-SMC: Proportional Integral Derivative Sliding Surface - Sliding Mode Control) thay cho mặt trượt cổ điển như Hình 2.



Hình 2. Cấu trúc bộ PIDSS-SMC

Trong đó, $\Theta_d = [x_d \quad y_d \quad \theta_d]^T$ là quỹ đạo (vị trí của robot theo phương x và y) mong muốn của robot và $\Theta = [x_w \quad y_w \quad \theta]^T$ là quỹ đạo thực tế.

Sai số giữa Θ_d và Θ được định nghĩa như (2):

$$\mathbf{e}_{rr} = \Theta - \Theta_d \quad (2)$$

Lấy đạo hàm bậc một và hai của (2), ta được (3) và (4) :

$$\dot{\mathbf{e}}_{rr} = \dot{\Theta} - \dot{\Theta}_d \quad (3)$$

$$\ddot{\mathbf{e}}_{rr} = \ddot{\Theta} - \ddot{\Theta}_d \quad (4)$$

Mặt trượt PID cho điều khiển trượt được định nghĩa như (5) [17]:

$$\mathbf{S}_{PID} = \mathbf{K}_{Pro} \mathbf{e}_{rr} + \mathbf{K}_{Int} \int \mathbf{e}_{rr}(\tau) d\tau + \mathbf{K}_{Der} \dot{\mathbf{e}}_{rr} \quad (5)$$

Với $\mathbf{K}_{Pro} = \text{diag}(K_{Pro1}, K_{Pro2}, K_{Pro3}), K_{Pro1} > 0, K_{Pro2} > 0, K_{Pro3} > 0$

$\mathbf{K}_{Int} = \text{diag}(K_{Int1}, K_{Int2}, K_{Int3}), K_{Int1} > 0, K_{Int2} > 0, K_{Int3} > 0$

$\mathbf{K}_{Der} = \text{diag}(K_{Der1}, K_{Der2}, K_{Der3}), K_{Der1} > 0, K_{Der2} > 0, K_{Der3} > 0$.

Lấy đạo hàm của (5), ta được (6):

$$\dot{\mathbf{S}}_{PID} = \mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} \ddot{\mathbf{e}}_{rr} \quad (6)$$

Thế (4) vào (6), ta có (7):

$$\dot{\mathbf{S}}_{PID} = \mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\mathbf{P}_w \mathbf{X} + \mathbf{Q}_w \mathbf{U}_{con} + \mathbf{N}_f - \ddot{\Theta}_d) \quad (7)$$

Luật tiếp cận lũy thừa như (8) [18]:

$$\dot{\mathbf{S}}_{PID} = -\boldsymbol{\varepsilon} \text{sign}(\mathbf{S}_{PID}) - \boldsymbol{\mu} \mathbf{S}_{PID} \quad (8)$$

Trong đó, $\boldsymbol{\mu} = \text{diag}(\mu_1, \mu_2, \mu_3), \mu_1 > 0, \mu_2 > 0, \mu_3 > 0$, $\boldsymbol{\varepsilon} = \text{diag}(\varepsilon_1, \varepsilon_2, \varepsilon_3), \varepsilon_1 > 0, \varepsilon_2 > 0, \varepsilon_3 > 0$.

Lúc này, luật điều khiển PIDSS-SMC cho robot như (9):

$$\mathbf{U}_{con-PIDSS-SMC} = -(\mathbf{K}_{Der} \mathbf{Q}_w)^{-1} (\mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\mathbf{P}_w \mathbf{X} + \mathbf{N}_f - \ddot{\Theta}_d) + \boldsymbol{\varepsilon} \text{sign}(\mathbf{S}_{PID}) + \boldsymbol{\mu} \mathbf{S}_{PID}) \quad (9)$$

Theo luật điều khiển (9) thì $\Theta \rightarrow \Theta_d$ trong thời gian hữu hạn và hiện tượng chattering quanh mặt trượt được giảm.

Hàm Lyapunov như (10):

$$L = \frac{1}{2} \mathbf{S}_{PID}^2 \quad (10)$$

Lấy đạo hàm của (10), ta được (11):

$$\dot{L} = \mathbf{S}_{PID} \dot{\mathbf{S}}_{PID} \quad (11)$$

Thế (7) và (9) vào (11), ta được:

$$\begin{aligned} \dot{L} &= \mathbf{S}_{PID} (\mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\mathbf{P}_w \mathbf{X} + \mathbf{Q}_w \mathbf{U}_{con-PIDSS-SMC} + \mathbf{N}_f - \ddot{\Theta}_d)) \\ &= \mathbf{S}_{PID} \left(\mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} \left[\mathbf{P}_w \mathbf{X} + \mathbf{Q}_w \left[-(\mathbf{K}_{Der} \mathbf{Q}_w)^{-1} \left(\mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\mathbf{P}_w \mathbf{X} + \mathbf{N}_f - \ddot{\Theta}_d) + \boldsymbol{\varepsilon} \text{sign}(\mathbf{S}_{PID}) + \boldsymbol{\mu} \mathbf{S}_{PID} \right) \right] + \mathbf{N}_f - \ddot{\Theta}_d \right] \right) \\ &= \mathbf{S}_{PID} (-\boldsymbol{\varepsilon} \text{sign}(\mathbf{S}_{PID}) - \boldsymbol{\mu} \mathbf{S}_{PID}) = -\boldsymbol{\varepsilon} |\mathbf{S}_{PID}| - \boldsymbol{\mu} \mathbf{S}_{PID}^2 \leq 0 \end{aligned} \quad (12)$$

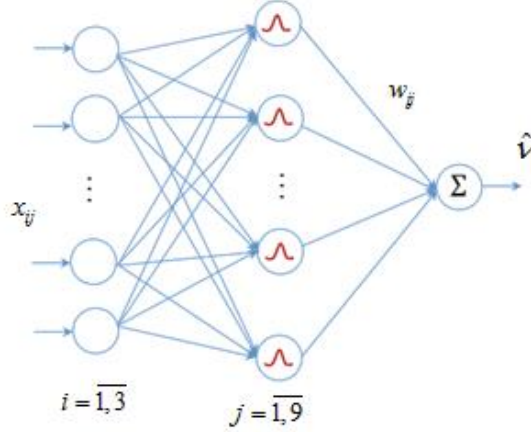
2.3. Ước lượng thành phần phi tuyến dựa vào mạng RBF-NN

RBF-NN là mạng nơ-ron một lớp ẩn [19] và có thể được coi là một ánh xạ: $R^r \rightarrow R^s$, bao gồm ba lớp khác nhau: lớp ngõ vào, lớp ẩn và lớp ngõ ra [20], [21].

RBF-NN có ưu điểm là thiết kế cấu trúc đơn giản, dễ huấn luyện, hội tụ nhanh, có thể phù hợp hiệu quả với bất kỳ hàm phi tuyến nào và không rơi vào lời giải tối ưu cục bộ [22] và có nhiều ứng dụng, bao gồm xấp xỉ hàm, phân loại và điều khiển hệ thống [23].

Cấu trúc [5-9-1] của mạng RBF-NN được sử dụng để xấp xỉ các thành phần $v_{i|i=1,2,3}$ trong ma trận \mathbf{P}_w của (9) như Hình 3.

Ma trận P_w trong luật điều khiển (9) chứa các thông số r , I_v và M của robot. Mạng RBF-NN sử dụng giải thuật Gradient Descent để cập nhật trực tuyến các giá trị trọng số. Mỗi mạng nơ-ron chứa 9 hàm Gauss và được mô tả như (13):



Hình 3. Mạng RBF-NN

$$h_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{c}_{ij}\|^2}{2b_{ij}^2}\right) \Big|_{i=1,3; j=1,9} \quad (13)$$

Trong đó

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} e_{rr}(1) & \dot{e}_{rr}(1) & \Theta_d(1) & \dot{\Theta}_d(1) & \ddot{\Theta}_d(1) \\ e_{rr}(2) & \dot{e}_{rr}(2) & \Theta_d(2) & \dot{\Theta}_d(2) & \ddot{\Theta}_d(2) \\ e_{rr}(3) & \dot{e}_{rr}(3) & \Theta_d(3) & \dot{\Theta}_d(3) & \ddot{\Theta}_d(3) \end{bmatrix} \quad (14)$$

$$\mathbf{h}_{ij|i=1,2,3} = [h_{i1} \ h_{i2} \ h_{i3} \ h_{i4} \ h_{i5} \ h_{i6} \ h_{i7} \ h_{i8} \ h_{i9}] \quad (15)$$

$$\mathbf{w}_{ij|i=1,2,3} = [w_{i1} \ w_{i2} \ w_{i3} \ w_{i4} \ w_{i5} \ w_{i6} \ w_{i7} \ w_{i8} \ w_{i9}] \quad (16)$$

Ngõ ra của mạng RBF-NN như (17):

$$\hat{v}_i = \mathbf{w}_{ij}^T \mathbf{h}_{ij} \quad (17)$$

Hàm mục tiêu của mạng RBF-NN như (18):

$$E_i(t) = \frac{1}{2} (v_i(t) - \hat{v}_i(t))^2; \ i = 1, 2, 3 \quad (18)$$

Theo phương pháp Gradient Descent, các giá trị w của mạng RBF-NN được cập nhật như (19), (20)

$$\Delta w_j(t) = -\gamma \frac{\partial E}{\partial w_j} = \lambda (v_i(t) - \hat{v}_i(t)) h_j \quad (19)$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha (w_j(t-1) - w_j(t-2)) \quad (20)$$

Trong đó, $\gamma \in (0,1)$ là tốc độ học và $\alpha \in (0,1)$ là yếu tố động lượng.

Lúc này, ma trận xấp xỉ xi của P_w như (21):

$$\hat{\mathbf{P}}_w = \begin{bmatrix} \mathbf{w}_{1j}^T \mathbf{h}_{1j} & -\mathbf{w}_{2j}^T \mathbf{h}_{2j} \dot{\theta} & 0 \\ \mathbf{w}_{2j}^T \mathbf{h}_{2j} \dot{\theta} & \mathbf{w}_{1j}^T \mathbf{h}_{1j} & 0 \\ 0 & 0 & \mathbf{w}_{3j}^T \mathbf{h}_{3j} \end{bmatrix} \quad (21)$$

Luật PIDSS-SMC sử dụng mạng RBF-NN (PIDSS-SMC-RBF-NN) cho Mobile robot như (22):

$$\mathbf{U}_{con-PIDSS-SMC-RBF-NN} = -(\mathbf{K}_{Der} \mathbf{Q}_w)^{-1} \left(\mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\hat{\mathbf{P}}_w \mathbf{X} + \mathbf{N}_f - \ddot{\Theta}_d) + \varepsilon \text{sign}(\mathbf{S}_{PID}) + \mu \mathbf{S}_{PID} \right) \quad (22)$$

Để chứng minh tính ổn định, hàm Lyapunov được định nghĩa như (23):

$$L = \frac{1}{2} \mathbf{S}_{PID}^2 \quad (23)$$

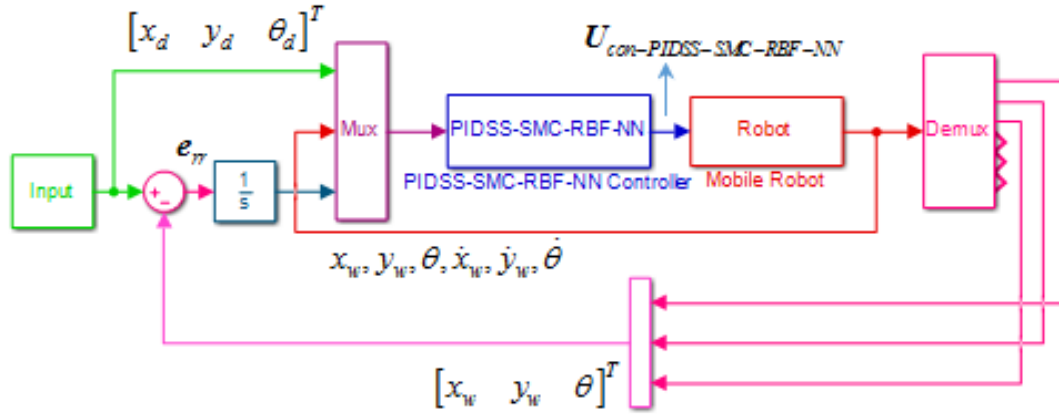
Lấy đạo hàm của (23), ta được (24):

$$\begin{aligned} \dot{L} &= \mathbf{S}_{PID} \dot{\mathbf{S}}_{PID} \\ &= \mathbf{S}_{PID} \left\{ \mathbf{K}_{Pro} \dot{\mathbf{e}}_{rr} + \mathbf{K}_{Int} \mathbf{e}_{rr} + \mathbf{K}_{Der} (\hat{\mathbf{P}}_w \mathbf{X} + \mathbf{Q}_w \mathbf{U}_{con-PIDSS-SMC-RBF-NN} + \mathbf{N}_f - \ddot{\Theta}_d) \right\} \\ &= -\varepsilon |\mathbf{S}_{PID}| - \mu \mathbf{S}_{PID}^2 \leq 0 \end{aligned} \quad (24)$$

Ta có thể thấy rằng $\mathbf{S}_{PID} \rightarrow \mathbf{0}$ khi $t \rightarrow 0$. Vì thế, $\mathbf{e}_{rr}, \dot{\mathbf{e}}_{rr} \rightarrow \mathbf{0}$ khi $t \rightarrow 0$.

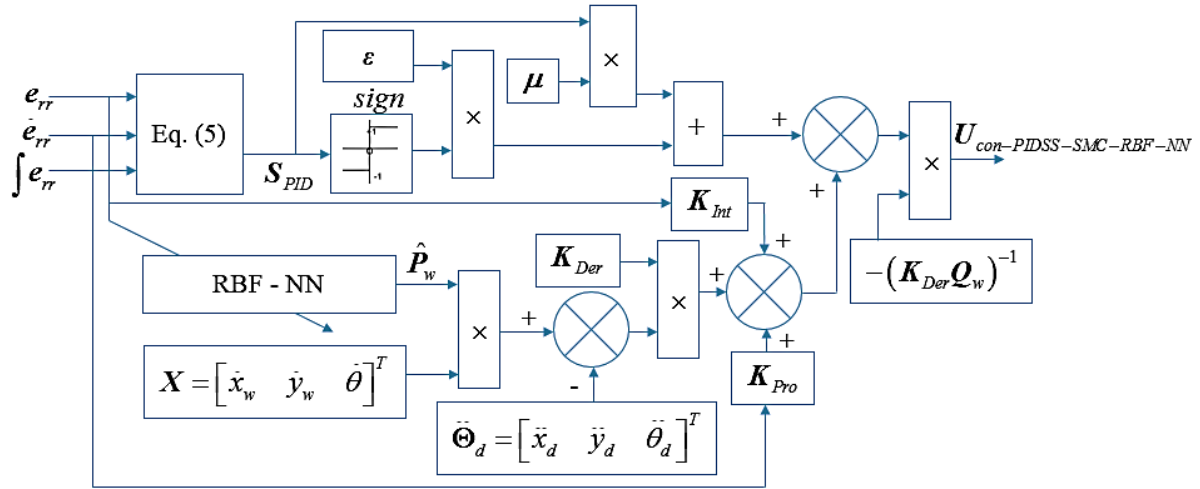
3. Kết quả và bàn luận

Bộ điều khiển PIDSS-SMC-RBF-NN trong môi trường MATLAB/Simulink cho Mobile robot như Hình 4.



Hình 4. Bộ điều khiển PIDSS-SMC-RBF-NN trong môi trường MATLAB/Simulink

Các khối Robot và PIDSS-SMC-RBF-NN được thực hiện bằng S-Function trong MATLAB/Simulink. Hình 5 trình bày chi tiết khối PIDSS-SMC-RBF-NN.



Hình 5. Chi tiết khối PIDSS-SMC-RBF-NN

Các thông số của robot được sử dụng trong mô phỏng như Bảng 1 sau:

Bảng 1. Các thông số của robot [13]

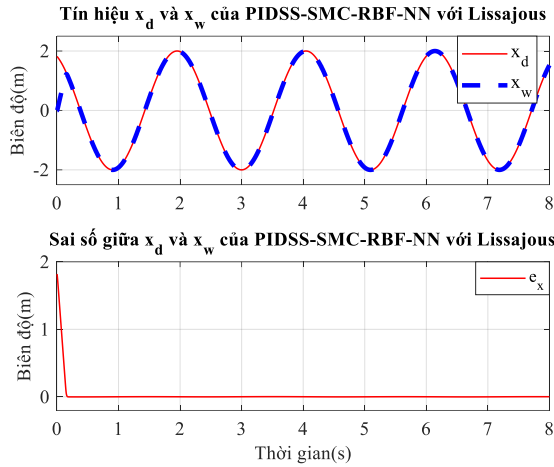
Ký hiệu	Ý nghĩa	Giá trị	Đơn vị
I_v	Mô-men quán tính của robot	11,25	kgm^2
M	Khối lượng của robot	9,4	kg
L	Khoảng cách từ mỗi bánh xe đến tâm robot	0,178	m
k	Hệ số truyền động	0,448	
c	Hệ số ma sát nhớt	0,1889	kgm^2 / s
I_w	Mô-men quán tính của bánh xe	0,02108	kgm^2
r	Bán kính mỗi bánh xe	0,0245	m

Các thông số mô phỏng bộ điều khiển PIDSS-SMC-RBF-NN như sau: $K_{Pro} = diag([2, 2, 2])$, $K_{Int} = diag([0,05, 0,05, 0,05])$, $K_{Der} = diag([0,015, 0,015, 0,015])$, $\epsilon = diag([15, 15, 8])$, $\mu = diag([200, 200, 200])$.

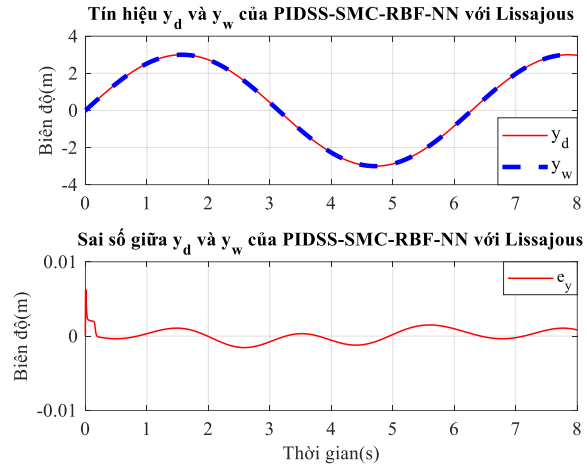
Tín hiệu x_w hội tụ về x_d với đường cong Lissajous [24] của bộ điều khiển PIDSS-SMC-RBF-NN như Hình 6 và Hình 7 đối với y_w và y_d . Quan sát Hình 6 ta thấy rằng x_w hội tụ về x_d với $t_{r-x} = 0,0832s$, $t_{ss-x} = 0,1309s$, e_{ss-x} tiến về 0 và $POT_x = 0,0042\%$; và từ Hình 7 y_w hội tụ về y_d với $t_{r-y} = 0,0764s$, $t_{ss-y} = 0,1226s$, e_{ss-y} tiến về 0 và $POT_y = 0,0055\%$ như Bảng 2. Giá trị ADD và MSE [25] theo phương x là $2,2298e-07$ và $3,9779e-10$, theo phương y là $5,3586e-09$ và $2,2975e-13$.

$$ADD = \frac{1}{N} \sum_{t=1}^N |y(t) - y_d(t)| \quad (25)$$

$$MSE = \frac{1}{N} \sum_{t=1}^N (y(t) - y_d(t))^2 \quad (26)$$



Hình 6. Tín hiệu của x_d và x_w bộ PIDSS-SMC-RBF-NN với đường cong Lissajous



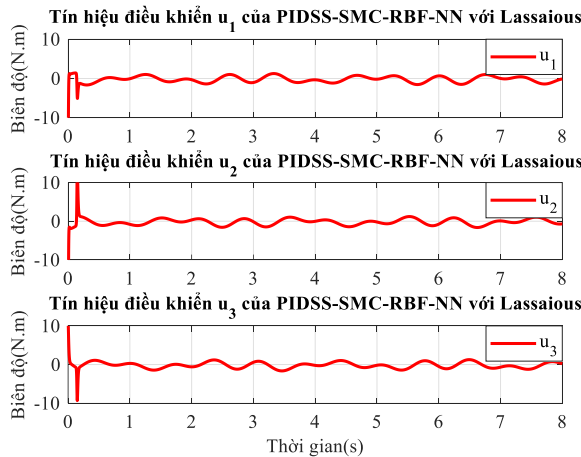
Hình 7. Tín hiệu của y_d và y_w bộ PIDSS-SMC-RBF-NN với đường cong Lissajous

Bảng 2. Các thông số kỹ thuật đạt được của bộ điều khiển PIDSS-SMC-RBF-NN

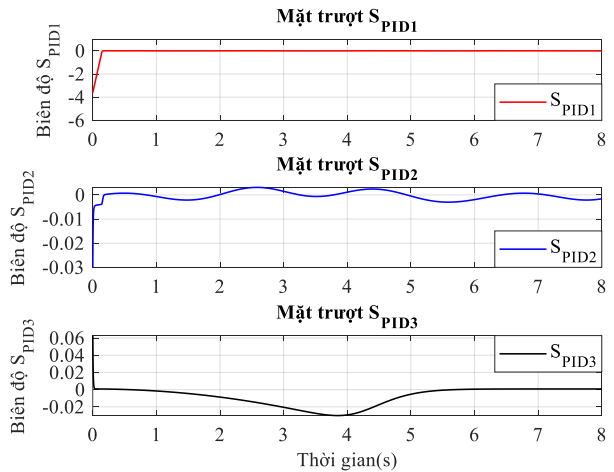
Chỉ tiêu chất lượng	t_r (s)	t_{ss} (s)	e_{ss} (m)	POT (%)
x_w	0,0832	0,1309	0	0,0042
y_w	0,0764	0,1226	0	0,0055

Trong đó, t_r là thời gian tăng, t_{ss} là thời gian xác lập, e_{ss} là sai số xác lập và POT là độ vọt lố.

Hình 8 trình bày $U_{con-PIDSS-SMC-RBF-NN} = [u_1 \ u_2 \ u_3]^T$ của bộ điều khiển PIDSS-SMC-RBF-NN. Các tín hiệu điều khiển này có biên độ nhỏ và tần số dao động thấp. Kết quả này cho thấy hiệu quả của PIDSS trong việc giảm chattering trong điều khiển trượt.

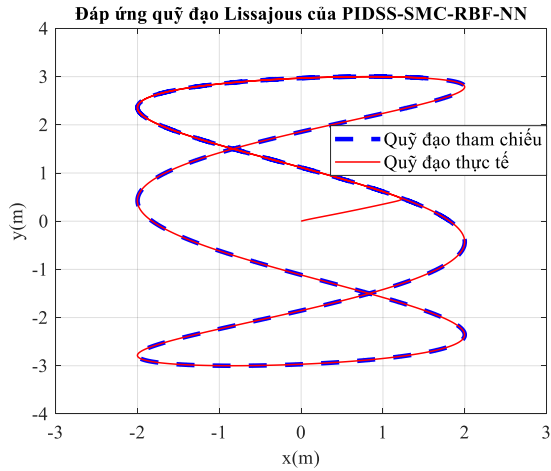


Hình 8. $U_{con-PIDSS-SMC-RBF-NN} = [u_1 \ u_2 \ u_3]^T$ của bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Lissajous

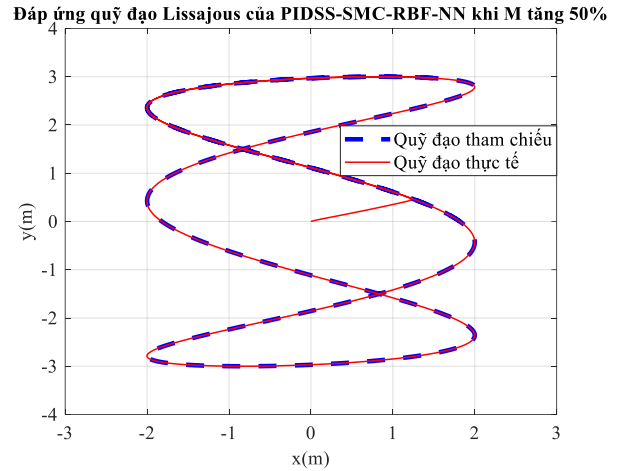


Hình 9. $S_{PID} = [S_{PID1} \ S_{PID2} \ S_{PID3}]^T$ của bộ điều khiển PIDSS-SMC-RBF-NN

Hình 9 trình bày các mặt trượt của bộ điều khiển PIDSS-SMC-RBF-NN. Mặt trượt này lúc khởi động theo giá trị hệ số trượt. Sau đó, các mặt trượt nhanh chóng đạt đến điểm hội tụ (điểm ổn định) và tiếp tục trượt quanh giá trị 0.



Hình 10. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Lissajous

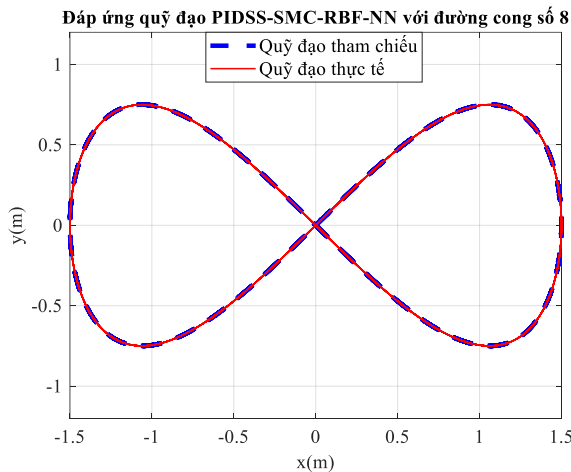


Hình 11. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Lissajous khi M tăng 50%

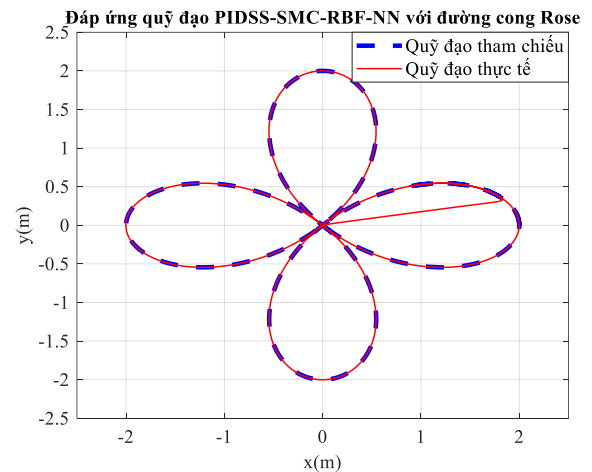
Ngõ ra thực tế theo phương x và y với đường cong Lissajous của bộ điều khiển PIDSS-SMC-RBF-NN như Hình 10. Ngõ ra thực tế này của robot tiến về ngõ vào chuẩn với sai lệch hội tụ về 0.

Hình 11 trình bày ngõ ra thực tế theo phương x và y với đường cong Lissajous của bộ điều khiển PIDSS-SMC-RBF-NN khi giá trị M tăng thêm 50% so với giá trị ban đầu. Ngõ ra thực tế của robot vẫn tiến về ngõ vào đặt với sai lệch hội tụ về 0.

Hình 12 và Hình 13 trình bày ngõ ra thực tế theo phương x và y của bộ điều khiển PIDSS-SMC-RBF-NN với đường cong hình số 8 và đường cong Rose tương ứng. Các ngõ ra này của robot vẫn hội tụ về ngõ vào đặt.

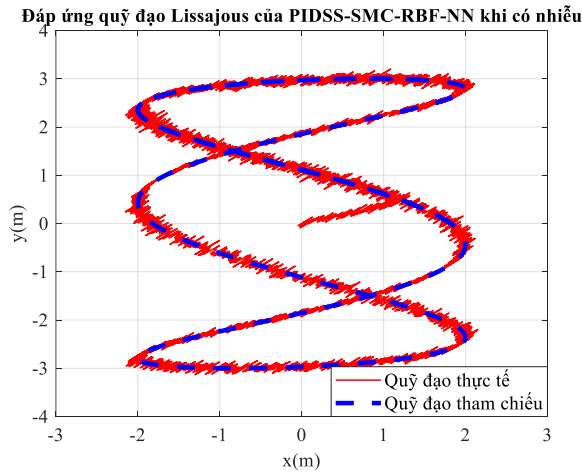


Hình 12. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong số 8

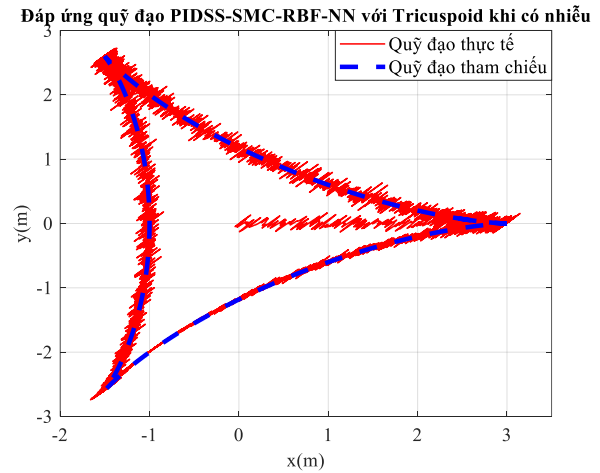


Hình 13. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Rose

Hình 14 và Hình 15 trình bày ngõ ra thực tế theo phương x và y của bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Lissajous và Tricuspid tương ứng trong trường hợp nhiễu tác động vào ngõ ra của robot (giả sử nhiễu cảm biến có công suất là 0,0005w, thời gian lấy mẫu là 0,0001s). Ngõ ra thực tế theo phương x và y của robot vẫn hội tụ về ngõ vào cho trước.



Hình 14. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Lissajous khi có nhiễu



Hình 15. Ngõ ra thực tế theo phương x và y bộ điều khiển PIDSS-SMC-RBF-NN với đường cong Tricuspid khi có nhiễu

Các kết quả đạt được như trên đã cho thấy tính bền vững, sự phù hợp và hiệu quả của phương pháp điều khiển PIDSS-SMC-RBF-NN cho Mobile robot.

4. Kết luận


Bài báo đã ứng dụng PIDSS và mạng RBF-NN trong điều khiển bám quỹ đạo cho Mobile robot. Các kết quả mô phỏng trong môi trường MATLAB/Simulink cho thấy hiệu quả của bộ điều khiển đề xuất, tín hiệu thực tế của x_w hội tụ về x_d với $t_{r-x} = 0,0832s$, $t_{ss-x} = 0,1309s$, e_{ss-x} tiến về 0 và $POT_x = 0,0042\%$; và từ Hình 7 y_w hội tụ về y_d với $t_{r-y} = 0,0764s$, $t_{ss-y} = 0,1226s$, e_{ss-y} tiến về 0 và $POT_y = 0,0055\%$. Giá trị ADD và MSE của phương pháp điều khiển đề xuất đạt được theo phương x là $2,2298e-07$ và $3,9779e-10$, theo phương y là $5,3586e-09$ và $2,2975e-13$ và hiện tượng chattering được giảm. Kết quả khảo sát với trường hợp ngõ vào chuẩn thay đổi và có nhiễu tác động ở ngõ ra của robot bộ điều khiển đề xuất đã cho thấy tính bền vững, sự phù hợp và hiệu quả của bộ điều khiển này trong ứng dụng điều khiển bám quỹ đạo robot. Trong thời gian tới, nhóm tác giả sẽ nghiên cứu thuật toán di truyền hoặc thuật toán tối ưu hóa bầy đàn hoặc thuật toán tối ưu hóa cá voi để xác định số lượng nơ-ron phù hợp trong lớp ẩn của mạng RBF-NN và thực nghiệm với mô hình robot thực tế.

TÀI LIỆU THAM KHẢO


- [1] C. Ren, C. Li, L. Hu, X. Li, and S. Ma, "Adaptive model predictive control for an omnidirectional mobile robot with friction compensation and incremental input constraints," *Trans. Inst. Meas. Control*, vol. 44, no. 4, pp. 835–847, 2022, doi: 10.1177/01423312211021321.
- [2] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, Anchorage, AK, USA: IEEE, 1998, pp. 69–73, doi: 10.1109/ICEC.1998.699146.
- [3] Y. Liu, J. J. Zhu, R. L. Williams, and J. Wu, "Omni-directional mobile robot controller based on trajectory linearization," *Robot. Auton. Syst.*, vol. 56, no. 5, pp. 461–479, May 2008, doi: 10.1016/j.robot.2007.08.007.
- [4] Q. Xu, J. Kan, S. Chen, and S. Yan, "Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink," *Int. J. Control Autom.*, vol. 7, no. 8, pp. 233–244, Aug. 2014, doi: 10.14257/ijca.2014.7.8.20.
- [5] L. Ovalle, H. Ríos, M. Llama, V. Santibáñez, and A. Dzul, "Omnidirectional mobile robot robust tracking: Sliding-mode output-based control approaches," *Control Eng. Pract.*, vol. 85, no. 4, pp. 50–58, Apr. 2019, doi: 10.1016/j.conengprac.2019.01.002.
- [6] D. T. Nguyen, C. C. Tran, and H. D. Le, "Modeling and Control of Three Wheeled Omni-Directional Mobile Robot," in *Proc. the 8th Vietnam Conference on Mechatronics*, 2016, pp. 517–523.
- [7] H. D. Le, D. T. Nguyen, and C. C. Tran, "Experiment on a Omni-Directional Mobile Robot Using RBF-PD Supervisory Controller," *Measurement, Control and Automation*, vol. 17, pp. 51–55, 2016.
- [8] T. T. Pham, D. V. Huong, C. N. Nguyen, and M. T. Le, "Comparison of SMC and RBF-SMC on mobile robot control system," *16th ASIA Marit. Fish. Univ. FORUM*, 2017, pp. 325–339.
- [9] W. Xiao, G. Wang, J. Tian, and L. Yuan, "A novel adaptive robust control for trajectory tracking of mobile robot with uncertainties," *J. Vib. Control*, pp. 563–574, 2023, doi: 10.1177/10775463231161847.
- [10] G. D. S. Lima, V. R. F. Moreira, and W. M. Bessa, "Accurate trajectory tracking control with adaptive neural networks for omnidirectional mobile robot subject to unmodeled dynamics," *J. Braz. Soc. Mech. Sci. Eng.*, vol. 45, no. 1, pp. 1–11, 2023, doi: 10.1007/s40430-022-03969-y.

- [11] X. Feng and C. Wang, "Adaptive neural network tracking control of an omnidirectional mobile robot," *Proc. Inst. Mech. Eng. Part J. Syst. Control Eng.*, vol. 237, no. 3, pp. 375–387, 2023, doi: 10.1177/09596518221135904.
- [12] L. T. Hoan, T. Dong, and V. V. Thong, "Adaptive Sliding Mode Control for Three-Wheel Omnidirectional Mobile Robot," *Int. J. Eng. Trends Technol.*, vol. 71, no. 5, pp. 9–17, 2023, doi: 10.14445/22315381/IJETT-V71I5P202.
- [13] A. Mehmood, I. U. H. Shaikh, and A. Ali, "Application of Deep Reinforcement Learning for Tracking Control of 3WD Omnidirectional Mobile Robot," *Inf. Technol. Control*, vol. 50, no. 3, pp. 507–521, Sep. 2021, doi: 10.5755/j01.itc.50.3.25979.
- [14] W. Zheng and Y. Jia, "Trajectory Tracking Control for Omnidirectional Mobile Robots with Full-State Constraints," in *Proceedings of 2017 Chinese Intelligent Automation Conference*, vol. 458, Z. Deng, Ed., in Lecture Notes in Electrical Engineering, vol. 458, Singapore: Springer Singapore, 2018, pp. 605–612, doi: 10.1007/978-981-10-6445-6_66.
- [15] C. G. Yun, Y. C. Sin, H. R. Ri, and K. N. Jo, "Trajectory tracking control of a three-wheeled omnidirectional mobile robot using disturbance estimation compensator by RBF neural network," *J. Braz. Soc. Mech. Sci. Eng.*, vol. 45, no. 8, p. 432, 2023, doi: 10.1007/s40430-023-04340-5.
- [16] K. Watanabe, "Control of an omnidirectional mobile robot," *1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98* (Cat. No.98EX111), Adelaide, SA, Australia, 1998, pp. 51–60, doi: 10.1109/KES.1998.725827.
- [17] F. Loucif and S. Kechida, "Optimization of sliding mode control with PID surface for robot manipulator by Evolutionary Algorithms," *Open Comput. Sci.*, vol. 10, no. 1, pp. 396–407, 2020, doi: 10.1515/comp-2020-0144.
- [18] J. Liu, *Sliding Mode Control Using MATLAB*. Academic Press, 2017.
- [19] H. Li and S. Huang, "Research on the Prediction Method of Stock Price Based on RBF Neural Network Optimization Algorithm," *E3S Web Conf.*, vol. 235, pp. 1–5, 2021, doi: 10.1051/e3sconf/202123503088.
- [20] H. Wang, Y. Zhao, J. Pei, D. Zeng, and M. Liu, "Non-negative Radial Basis Function Neural Network in Polynomial Feature Space," *J. Phys. Conf. Ser.*, vol. 1168, no. 6, pp. 1–8, Feb. 2019, doi: 10.1088/1742-6596/1168/6/062005.
- [21] A. Lemita, S. Boulahbel, and S. Kahla, "Gradient Descent Optimization Control of an Activated Sludge Process based on Radial Basis Function Neural Network," *Eng. Technol. Appl. Sci. Res.*, vol. 10, no. 4, pp. 6080–6086, Aug. 2020, doi: 10.48084/etasr.3714.
- [22] A. İ. Kaya, M. İlkuçar, and A. ÇiFci, "Use of Radial Basis Function Neural Network in Estimating Wood Composite Materials According to Mechanical and Physical Properties," *Erzincan Üniversitesi Fen Bilim. Enstitüsü Derg.*, vol. 12, no. 1, pp. 116–123, Mar. 2019, DOI: 10.18185/erzifbed.428763.
- [23] A. Lemita, S. Boulahbel, S. Kahla, and M. Sedraoui, "Auto-Control Technique Using Gradient Method Based on Radial Basis Function Neural Networks to Control of an Activated Sludge Process of Wastewater Treatment," *J. Eur. Systèmes Autom.*, vol. 53, no. 5, pp. 671–679, 2020, doi: 10.18280/jesa.530510.
- [24] C. Tran, "Catalog of curves-part-2," 2012, [Online]. Available: <http://cohtran.blogspot.com/2012/04/catalog-of-curves-part-2>.
- [25] I. Mukherjee and S. Routroy, "Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2397–2407, Feb. 2012, doi: 10.1016/j.eswa.2011.08.087.




Tran Trung Hieu received BSc degree in HCMC University of Technical and Education in 2002. Currently, he works at the Ly Tu Trong College Ho Chi Minh City and is the master student of the Vinh Long University of Technical Education. His research interests include robot and modern control engineering. Email: trantrunghieu.ltt@gmail.com. ORCID:  <https://orcid.org/0009-0002-9512-4745>




Nguyen Viet Trung received BSc degree in Vinh Long University of Technical Education in 2018, and MSc degree in Vinh Long University of Technical Education in 2021. Currently, he works at the Faculty of Electrical and Electronics Engineering, Vinh Long University of Technical Education. His research interests include electric motor, Microcontrollers, Embedded System and Artificial Neural Network. Email: trungnv@vlute.edu.vn. ORCID:  <https://orcid.org/0009-0004-2887-361X>



Tran Thi Thuy Trang received BSc degree in Vinh Long University of Technical Education in 2019, and MSc degree in Vinh Long University of Technical Education in 2022. Currently, she works at the Faculty of Electrical and Electronics Engineering, Vinh Long University of Technical Education. Her research interests include robot, Microcontrollers and modern control engineering. Email: trangttt@vlute.edu.vn. ORCID:  <https://orcid.org/0009-0005-6688-6785>



Pham Thanh Tung received BSc degree in EE at Mekong University in 2004, and MSc degree in Automation at Ho Chi Minh City University of Transportation in 2010. The degree of Ph.D. was award by the Ho Chi Minh City University of Transport, Vietnam, in 2019. Currently, he works at the Faculty of Electrical and Electronics Engineering, Vinh Long University of Technical Education. His research interests include robot, intelligent and modern control engineering. Email: tungpt@vlute.edu.vn. ORCID:  <https://orcid.org/0000-0001-7437-9541>