

AN IMPROVED SLIDING SURFACE AND ARTIFICIAL NEURAL NETWORK WITH AN APPLICATION FOR ROBOT CONTROL

Tran Trung Hieu, Pham Thanh Tung*

Vinh Long University of Technology Education

ARTICLE INFO	ABSTRACT
Received: 13/12/2023	Mobile robots are autonomous devices that capable of moving on themselves, with applications ranging from surveillance and warehouse logistics to healthcare and planetary exploration. Precise trajectory tracking control is a key component in robotic applications. This research applies an improved sliding surface and artificial neural network for Mobile robot. The improved sliding surface combined with exponential approach law and hyperbolic tangent function are used to reduce the chattering phenomenon in the sliding mode control. The nonlinear components in the sliding mode control law are estimated using an artificial neural network. The weights of this neural network are updated online using the gradient descent algorithm. Lyapunov theory is used to prove the stability of the system. Simulation results in MATLAB/Simulink show the effectiveness of the proposed method with the rising time achieves 0.071(s), the overshoot is 0.004(%), the steady-state error converges to zero, and the settling time is about 0.0978(s) in x-coordinate and 0.0646(s), 0.0042(%), 0(m) và 0.0902(s) in y-coordinate, respectively; the chattering phenomena has small amplitude and low oscillation frequency.
Revised: 29/01/2024	
Published: 30/01/2024	
KEYWORDS	
Sliding mode control	
Mobile robot	
Improved sliding surface	
Artificial neural network	
MATLAB/Simulink	

MẶT TRƯỢT CẢI TIẾN VÀ MẠNG NƠ-RON NHÂN TẠO VỚI ỨNG DỤNG CHO ĐIỀU KHIỂN ROBOT

Trần Trung Hiếu, Phạm Thanh Tùng*

Trường Đại học Sư phạm Kỹ thuật Vĩnh Long

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 13/12/2023	Mobile robot là thiết bị tự động có khả năng tự di chuyển với các ứng dụng từ hoạt động giám sát và hậu cần kho hàng đến dịch vụ chăm sóc sức khỏe và khám phá hành tinh. Điều khiển bám quỹ đạo chính xác là thành phần quan trọng trong các ứng dụng của robot. Nghiên cứu này ứng dụng mặt trượt cải tiến và mạng nơ-ron nhân tạo cho Mobile robot. Mặt trượt cải tiến kết hợp với luật tiếp cận hàm mũ và hàm Hyperbolic tangent được sử dụng để giảm hiện tượng chattering trong điều khiển trượt. Các thành phần phi tuyến trong luật điều khiển trượt được ước lượng bằng mạng nơ-ron nhân tạo. Các trọng số của mạng nơ-ron được cập nhật trực tuyến bằng giải thuật gradient descent. Tính ổn định của hệ thống được chứng minh bằng lý thuyết Lyapunov. Các kết quả mô phỏng với MATLAB/Simulink cho thấy hiệu quả của phương pháp đề xuất với thời gian tăng đạt 0,071(s), độ quá điều chỉnh là 0,004(%), sai số xác lập về 0(m), thời gian xác lập là 0,0978(s) theo trục x và 0,0646(s), 0,0042(%), 0(m) và 0,0902(s) theo trục y, tương ứng; hiện tượng chattering có biên độ nhỏ và tần số dao động thấp.
Ngày hoàn thiện: 29/01/2024	
Ngày đăng: 30/01/2024	
TỪ KHÓA	
Điều khiển trượt	
Mobile robot	
Mặt trượt cải tiến	
Mạng nơ-ron nhân tạo	
MATLAB/Simulink	

DOI: <https://doi.org/10.34238/tnu-jst.9400>

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

1. Giới thiệu

Mobile robot là thiết bị tự động có khả năng tự di chuyển và tương tác với môi trường [1]. Điều khiển bám quỹ đạo chính xác là thành phần quan trọng trong các ứng dụng của robot, tiêu biểu như điều khiển tuyến tính hóa quỹ đạo dựa vào mô hình động học của robot trong [2]; điều khiển trượt thích nghi được phát triển trong [3]; điều khiển mờ trong [4] đã đạt được mục tiêu và duy trì ở đó sau 2,23 giây, sai số tối đa nhỏ hơn 1 mm; điều khiển PI được trình bày trong [5]; phương pháp PD+ đã được phát triển trong [6], giá trị của ISI của bộ điều khiển là 8705235,5, chỉ số RMSE lần lượt là 46,41 và 6,56; điều khiển trượt động PI được đề xuất trong [7]; điều khiển phi tuyến thích nghi được thiết kế và mô phỏng trong [8] đã cho thấy quỹ đạo vòng tròn của robot đạt được mong muốn trong thời gian 0,5 giây mà không có vọt lố, sai số ở trạng thái ổn định là không đáng kể; điều khiển dự báo mô hình thích nghi (MPC) với ma sát đã được phát triển trong [9], kết quả thực nghiệm với quỹ đạo hình số 8 cho thấy IAE_{xy} và IAE_{θ} của MPC lần lượt là 1,517 m và 1,2907 rad, MAE_{xy} và MAE_{θ} lần lượt là 0,0997 m và 0,0655 rad.

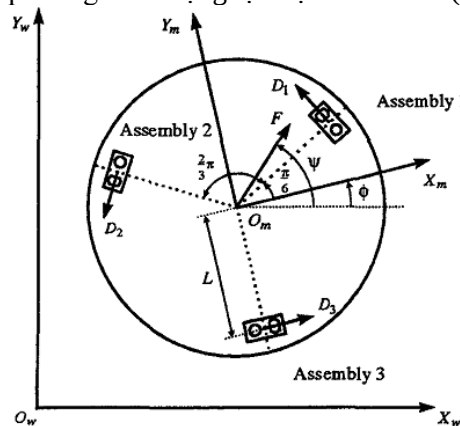
Bộ điều khiển trượt cho robot này đã được thực hiện trong các nghiên cứu [3] và [7], tuy nhiên, vấn đề chattering xảy ra trong luật điều khiển trượt chưa được đề cập trong các công bố này. Đây là hạn chế của bộ điều khiển trượt bên cạnh ưu điểm đáp ứng nhanh và bền vững với sự thay đổi của nhiễu cũng như các thông số của đối tượng. Để giải quyết khó khăn trên, nghiên cứu này đề xuất sử dụng mặt trượt cải tiến dựa vào mạng nơ-ron nhân tạo (mạng nơ-ron RBF: Radial Basis Function) để điều khiển bám quỹ đạo robot. Mặt trượt cải tiến được thiết kế thay cho mặt trượt cổ điển kết hợp với luật tiếp cận hàm mũ và hàm Hyperbolic tangent ($\tanh$) góp phần giảm hiện tượng chattering quanh mặt trượt. Mạng RBF được sử dụng để ước lượng các thành phần phi tuyến trong luật điều khiển trượt được tính toán dựa trên lý thuyết ổn định Lyapunov.

Bài báo này được tổ chức gồm 4 phần: phần 2 trình bày ứng dụng mặt trượt cải tiến và mạng nơ-ron nhân tạo cho robot, kết quả và bàn luận được trình bày trong phần 3 và phần 4 là kết luận.

2. Phương pháp ứng dụng mặt trượt cải tiến và mạng nơ-ron nhân tạo cho robot

2.1. Phương trình toán học robot

Mô hình robot như Hình 1 và phương trình động lực học robot như (1) [10] sau:



Hình 1. Mô hình robot [10]

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} a_1 & -a_2\dot{\phi} & 0 \\ a_2\dot{\phi} & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi \\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi \\ b_2 & b_2 & b_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} D_{fx} \\ D_{fy} \\ D_{f\phi} \end{bmatrix} \quad (1)$$

$$= A_w \beta + B_w U + D_f$$

Với $\mathbf{D}_f = [D_{\dot{x}} \quad D_{\dot{y}} \quad D_{\dot{\phi}}]^T$ là nhiễu hệ thống chưa biết.

Trong đó:

$$\mathbf{A}_w = \begin{bmatrix} a_1 & -a_2\dot{\phi} & 0 \\ a_2\dot{\phi} & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix}, \mathbf{B}_w = \begin{bmatrix} b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi \\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi \\ b_2 & b_2 & b_2 \end{bmatrix}, \mathbf{U} = [u_1 \quad u_2 \quad u_3]^T,$$

$$a_1 = \frac{-3c}{(3I_w + 2Mr^2)}; a'_2 = \frac{2Mr^2}{(3I_w + 2Mr^2)}; a_3 = \frac{-3cL^2}{(3I_wL^2 + I_vr^2)}; b_1 = \frac{kr}{(3I_w + 2Mr^2)}$$

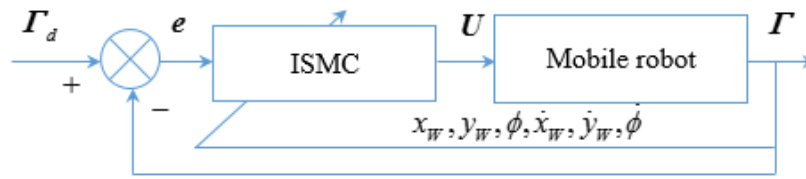
$$b_2 = \frac{krL}{(3I_w + I_vr^2)}; a_2 = 1 - a'_2 = \frac{3I_w}{(3I_w + 2Mr^2)}; \gamma_1 = -\sqrt{3}\sin\phi - \cos\phi;$$

$$\gamma_2 = \sqrt{3}\sin\phi - \cos\phi; \gamma_3 = \sqrt{3}\cos\phi - \sin\phi; \gamma_4 = -\sqrt{3}\cos\phi - \sin\phi$$

Trong đó: 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 hệ số tăng ích lái xe, u_i momen xoắn tác dụng lên xe.

2.2. Điều khiển robot sử dụng mặt trượt cải tiến

Sơ đồ cấu trúc bộ điều khiển trượt dựa vào mặt trượt cải tiến (ISMC) như Hình 2.



Hình 2. Sơ đồ cấu trúc bộ điều khiển trượt dựa vào mặt trượt cải tiến

Sự sai lệch giữa Γ và Γ_d như (2):

$$\mathbf{e} = \Gamma - \Gamma_d \quad (2)$$

Trong đó, $\Gamma = [x_w \quad y_w \quad \phi]^T$ là ngõ ra thực tế và $\Gamma_d = [x_d \quad y_d \quad \phi_d]^T$ là ngõ vào tham chiếu của robot.

Đạo hàm bậc 1 và 2 của (2), ta được (3) và (4):

$$\dot{\mathbf{e}} = \dot{\Gamma} - \dot{\Gamma}_d \quad (3)$$

$$\ddot{\mathbf{e}} = \ddot{\Gamma} - \ddot{\Gamma}_d = \begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\phi} \end{bmatrix} - \begin{bmatrix} \ddot{x}_d \\ \ddot{y}_d \\ \ddot{\phi}_d \end{bmatrix} = \dot{\mathbf{X}} - \ddot{\Gamma}_d = \mathbf{A}_w \mathbf{X} + \mathbf{B}_w \mathbf{U} + \mathbf{D}_f - \ddot{\Gamma}_d \quad (4)$$

Mặt trượt cải tiến như (5) [11].

$$\mathbf{s} = 2\dot{\mathbf{e}} + (\lambda + 2\alpha)\mathbf{e} + \alpha^2 \int_0^t \mathbf{e}(\tau) d\tau \quad (5)$$

Trong đó: $\lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$, $\lambda_{i|i=1,2,3} > 0$, $\alpha = \text{diag}(\alpha_1, \alpha_2, \alpha_3)$, $\alpha_{j|j=1,2,3} > 0$.

Thế (4) vào đạo hàm của (5), ta được (6):

$$\dot{s} = 2(A_w X + B_w U + D_f - \ddot{I}_d) + (\lambda + 2\alpha)\dot{e} + \alpha^2 e \quad (6)$$

Luật tiếp cận hàm mũ với hàm Hyperbolic tangent ($\tanh$) như (7) [12]:

$$\dot{s} = -\eta \tanh\left(\frac{s}{\varepsilon}\right) - \kappa s \quad (7)$$

Với $\eta = \text{diag}([\eta_x, \eta_y, \eta_\phi])$, $\kappa = \text{diag}([\kappa_x, \kappa_y, \kappa_\phi])$, $\varepsilon = \text{diag}([\varepsilon_x, \varepsilon_y, \varepsilon_\phi])$

và $\eta_x > 0, \eta_y > 0, \eta_\phi > 0$, $\kappa_x > 0, \kappa_y > 0, \kappa_\phi > 0$, $\varepsilon_x > 0, \varepsilon_y > 0, \varepsilon_\phi > 0$.

Luật điều khiển đề xuất cho robot như (8):

$$U = U_{ISMC-Tanh} = -\frac{1}{2} B_w^{-1} \left[2(A_w X + D_f - \ddot{I}_d) + (\lambda + 2\alpha)\dot{e} + \alpha^2 e + \eta \tanh\left(\frac{s}{\varepsilon}\right) + \kappa s \right] \quad (8)$$

Hàm Lyapunov như (9):

$$V = \frac{1}{2} s^2 \quad (9)$$

Lúc này,

$$\begin{aligned} \dot{V} &= s\dot{s} = s \left[2(A_w X + B_w U + D_f - \ddot{I}_d) + (\lambda + 2\alpha)\dot{e} + \alpha^2 e \right] \\ &= s \left[-\eta \tanh\left(\frac{s}{\varepsilon}\right) - \kappa s \right] = -\eta s \tanh\left(\frac{s}{\varepsilon}\right) - \kappa s^2 \end{aligned} \quad (10)$$

Theo [12, Bổ đề 1.2], ta có:

$$|s| - s \tanh\left(\frac{s}{\varepsilon}\right) \leq \mu \varepsilon \quad (11)$$

Thì

$$\eta |s| - \eta s \tanh\left(\frac{s}{\varepsilon}\right) \leq \eta \mu \varepsilon \quad (12)$$

Suy ra

$$-\eta s \tanh\left(\frac{s}{\varepsilon}\right) \leq -\eta |s| + \eta \mu \varepsilon \quad (13)$$

Vì thế:

$$\dot{V} = s\dot{s} \leq -\kappa s^2 - \eta |s| + \eta \mu \varepsilon \leq -\kappa s^2 + \eta \mu \varepsilon = -2\kappa V + \beta \quad (14)$$

Trong đó, $\beta = \eta \mu \varepsilon$

Sử dụng Bổ đề 1.3 trong [12], ta có:

$$\begin{aligned} \dot{V} &\leq e^{-2\kappa(t-t_0)} V(t_0) + \beta e^{-2\kappa t} \int_{t_0}^t e^{2\kappa \tau} d\tau = e^{-2\kappa(t-t_0)} V(t_0) + \frac{\beta e^{-2\kappa t}}{2\kappa} (e^{-2\kappa t} - e^{-2\kappa t_0}) \\ &= e^{-2\kappa(t-t_0)} V(t_0) + \frac{\beta}{2\kappa} (1 - e^{-2\kappa(t-t_0)}) = e^{-2\kappa(t-t_0)} V(t_0) + \frac{\eta \mu \varepsilon}{2\kappa} (1 - e^{-2\kappa(t-t_0)}) \end{aligned} \quad (15)$$

Vì thế

$$\lim_{t \rightarrow \infty} V(t) \leq \frac{\eta \mu \varepsilon}{2\kappa} \quad (16)$$

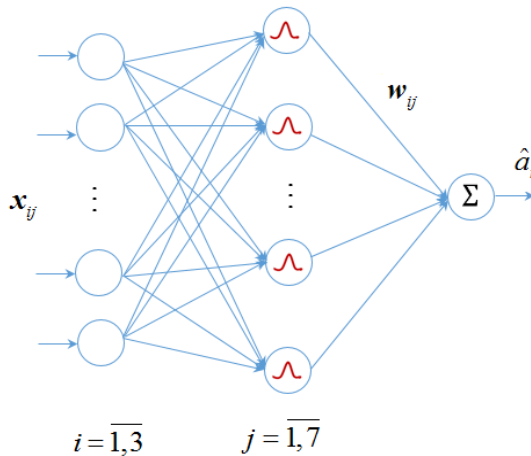
Theo bất đẳng thức (16), chúng ta có thể kết luận rằng sai số bám hội tụ tiệm cận và độ chính xác hội tụ phụ thuộc vào các giá trị $\varepsilon, \eta, \kappa$. Thêm vào đó, sai số $e(t)$ sẽ tiến về 0 dần theo

$s(t) \rightarrow 0$ khi $t \rightarrow \infty$. Vì thế $e(t), \dot{e}(t) \rightarrow 0$ khi $t \rightarrow \infty$.

2.3. Ước lượng các thành phần phi tuyến sử dụng mạng nơ-ron nhân tạo

Nghiên cứu này sử dụng mạng nơ-ron RBF để ước lượng các thành phần phi tuyến trong (8). Mạng nơ-ron RBF có nhiều ứng dụng, bao gồm ước lượng hàm, phân loại và điều khiển hệ thống với điểm mạnh 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 [13].

Cấu trúc [5-7-1] của mạng nơ-ron RBF được sử dụng để ước lượng các thành phần $a_{i|i=1,2,3}$ trong ma trận A_w của (8) như Hình 3.



Hình 3. Cấu trúc mạng nơ-ron RBF

Ma trận A_w trong luật điều khiển (8) chứa các thông số đặc trưng của robot như: bán kính của mỗi bánh xe (r); mô-men quán tính của robot (I_v) và khối lượng của robot (M). Mạng nơ-ron RBF sử dụng giải thuật Gradient Descent để cập nhật trực tuyến các giá trị trọng số. Mỗi mạng nơ-ron chứa 7 hàm Gauss và được mô tả như (17):

$$h_{ij} = \exp \left(- \frac{\| \mathbf{x}_i - \mathbf{c}_{ij} \|^2}{2b_{ij}^2} \right) \Bigg|_{i=\overline{1,3}; j=\overline{1,7}} \quad (17)$$

Trong đó

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} e(1) & \dot{e}(1) & \Gamma_d(1) & \dot{\Gamma}_d(1) & \ddot{\Gamma}_d(1) \\ e(2) & \dot{e}(2) & \Gamma_d(2) & \dot{\Gamma}_d(2) & \ddot{\Gamma}_d(2) \\ e(3) & \dot{e}(3) & \Gamma_d(3) & \dot{\Gamma}_d(3) & \ddot{\Gamma}_d(3) \end{bmatrix} \quad (18)$$

$$\mathbf{h}_{ij|i=1,2,3} = [h_{i1} \ h_{i2} \ h_{i3} \ h_{i4} \ h_{i5} \ h_{i6} \ h_{i7}] \quad (19)$$

$$\mathbf{w}_{ij|i=1,2,3} = [w_{i1} \ w_{i2} \ w_{i3} \ w_{i4} \ w_{i5} \ w_{i6} \ w_{i7}] \quad (20)$$

Ngõ ra của mạng nơ-ron như (21):

$$\hat{a}_i = \mathbf{w}_{ij}^T \mathbf{h}_{ij} \quad (21)$$

Hàm mục tiêu của mạng nơ-ron như (22):

$$E_i(t) = \frac{1}{2} (a_i(t) - \hat{a}_i(t))^2; i = 1, 2, 3 \quad (22)$$

Căn cứ theo phương pháp Gradient Descent, các giá trị trọng số của mạng được cập nhật như (23) và (24) [14]:

$$\Delta w_j(t) = -\mu \frac{\partial E}{\partial w_j} = \mu (a_i(t) - \hat{a}_i(t)) h_j \quad (23)$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \delta (w_j(t-1) - w_j(t-2)) \quad (24)$$

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

Lúc này, ma trận xấp xỉ của A_w như (25):

$$\hat{A}_w = \begin{bmatrix} \mathbf{w}_{1j}^T \mathbf{h}_{1j} & -\mathbf{w}_{2j}^T \mathbf{h}_{2j} \dot{\phi} & 0 \\ \mathbf{w}_{2j}^T \mathbf{h}_{2j} \dot{\phi} & \mathbf{w}_{1j}^T \mathbf{h}_{1j} & 0 \\ 0 & 0 & \mathbf{w}_{3j}^T \mathbf{h}_{3j} \end{bmatrix} \quad (25)$$

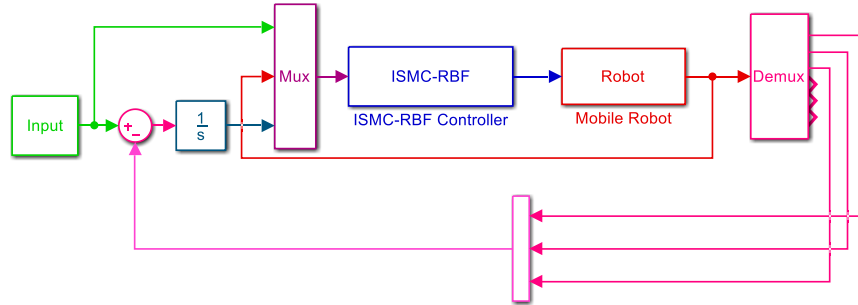
Luật điều khiển trượt dựa vào mặt trượt cải tiến và mạng nơ-ron RBF cho robot như (26):

$$\mathbf{U}_{ISMC-Tanh-RBF} = -\frac{1}{2} \mathbf{B}_w^{-1} \left[2(\hat{A}_w \mathbf{X} + \mathbf{D}_f - \ddot{\mathbf{I}}_d) + (\lambda + 2\alpha) \dot{\mathbf{e}} + \alpha^2 \mathbf{e} + \boldsymbol{\eta} \tanh\left(\frac{\mathbf{s}}{\boldsymbol{\varepsilon}}\right) + \boldsymbol{\kappa} \mathbf{s} \right] \quad (26)$$

Khi quỹ đạo thực tế của robot lệch khỏi quỹ đạo tham chiếu do tác động của các điều kiện như ma sát mặt đường, mô men quán tính thay đổi, v.v. thì sai số $\mathbf{e} = \mathbf{I} - \mathbf{I}_d$ sẽ thay đổi. Khi đó, mạng nơ-ron RBF sẽ tự động cập nhật dẫn đến thay đổi A_w sao cho các sai số có thể đạt giá trị cực tiểu. Bằng cách sử dụng mạng nơ-ron RBF trong luật (26), bộ điều khiển đề xuất có thể thích nghi với các điều kiện của robot.

3. Kết quả và bàn luận

Sơ đồ MATLAB/Simulink mô phỏng phương pháp đề xuất cho Mobile robot như Hình 4.

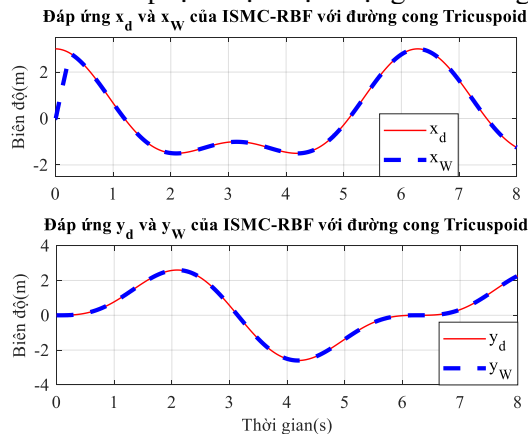


Hình 4. Sơ đồ MATLAB/Simulink mô phỏng bộ điều khiển ISMC-RBF

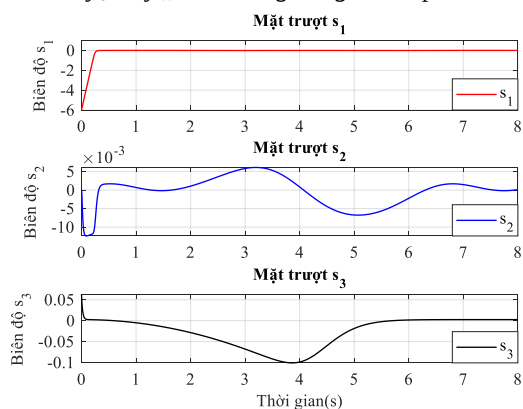
Các thông số của robot được sử dụng trong mô phỏng như sau: $I_v = 11,25(kgm^2)$, $M = 9,4(kg)$, $L = 0,178(m)$, $k = 0,448$, $c = 0,1889(kgm^2 / s)$, $I_w = 0,02108(kgm^2)$ và $r = 0,0245(m)$. Các thông số của bộ điều khiển đề xuất như sau: $\boldsymbol{\kappa} = diag([15,15,8])$, $\boldsymbol{\eta} = diag([200,200,200])$, $\boldsymbol{\varepsilon} = diag([0,15,0,15,0,15])$, $\boldsymbol{\lambda} = diag([15,15,15])$, $\boldsymbol{\alpha} = diag([12,12,12])$.

Đáp ứng của phương pháp ISMC-RBF giữa x_d và x_w , y_d và y_w với đường cong Tricuspidoid được trình bày như Hình 5. Quan sát Hình 5 ta thấy rằng x_w hội tụ về x_d với $t_{r_x} = 0,0710(s)$, $t_{ss_x} = 0,0978(s)$, $e_{ss_x} = 0(m)$ và $POT_x = 0,0040(%)$; và y_w hội tụ về y_d với $t_{r_y} = 0,0646(s)$, $t_{ss_y} = 0,0902(s)$, $e_{ss_y} = 0(m)$ và $POT_y = 0,0042(%)$.

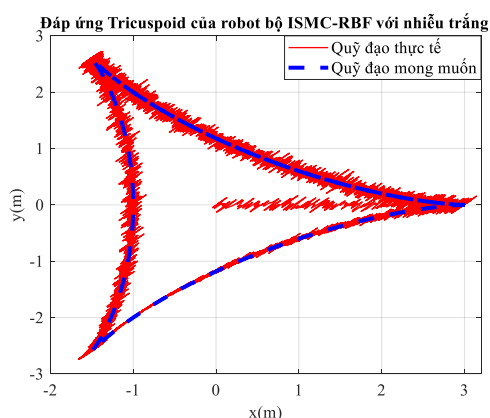
Hình 6 trình bày các tín hiệu điều khiển của bộ điều khiển ISMC-RBF. Các tín hiệu điều khiển này có biên độ nhỏ và tần số dao động thấp so với luật tiếp cận tốc độ hằng và hàm mũ với signum trong [15] có cùng thông số với robot này. Điều này chứng tỏ phương pháp điều khiển đề xuất đã khắc phục được hiện tượng chattering trong điều khiển trượt.



Hình 5. Đáp ứng của bộ ISMC-RBF giữa x_d và x_w ; y_d và y_w với đường cong Tricuspid

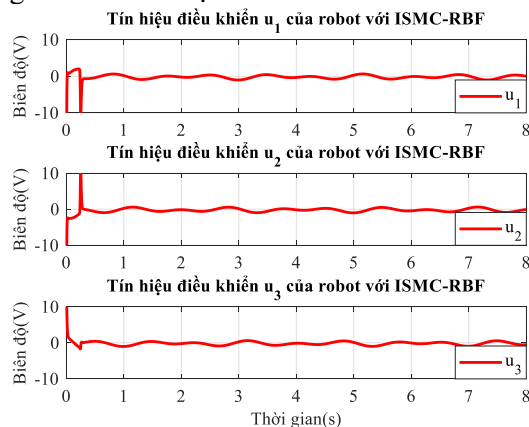


Hình 7. Các mặt trượt

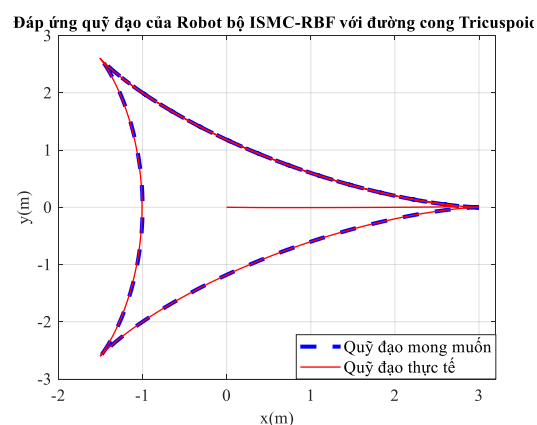


Hình 9. Đáp ứng quỹ đạo bộ ISMC-RBF của robot với đường cong Tricuspid khi có nhiễu

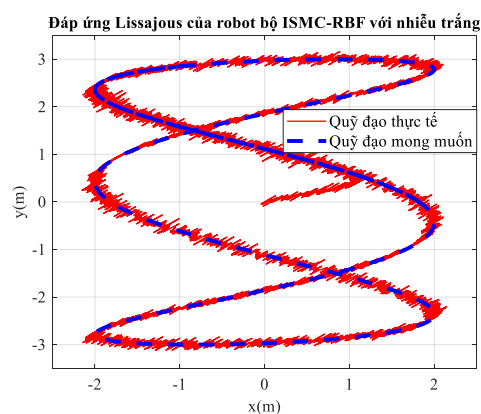
Hình 7 trình bày mặt trượt $s = [s_1 \ s_2 \ s_3]^T$ của bộ điều khiển đề xuất. Mặt trượt này lúc khởi động theo giá trị hệ số trượt. Sau đó, s nhanh chóng đạt đến điểm hội tụ và tiếp tục trượt quanh $s = 0$.



Hình 6. Tín hiệu điều khiển bộ ISMC-RBF



Hình 8. Đáp ứng quỹ đạo bộ ISMC-RBF của robot với đường cong Tricuspid



Hình 10. Đáp ứng quỹ đạo bộ ISMC-RBF của robot với đường cong Lissajous khi có nhiễu

Đáp ứng quỹ đạo Tricuspoind của bộ điều khiển ISMC-RBF được trình bày như Hình 8.

Hình 9 và Hình 10 trình bày các đáp ứng quỹ đạo của bộ điều khiển ISMC-RBF với đường cong Tricuspoind và Lissajous trong trường hợp có nhiễu tác động ở ngõ ra của robot (giả sử nhiễu cảm biến có công suất là 0,0005w, thời gian lấy mẫu là 0,0001s). Quỹ đạo thực tế của robot vẫn hội tụ về quỹ đạo tham chiếu trong thời gian hữu hạn với sai số hội tụ về 0.

4. Kết luận

Nghiên cứu đã ứng dụng mặt trượt cải tiến và mạng nơ-ron nhân tạo cho Mobile robot. Mặt trượt cải tiến kết hợp với luật tiếp cận hàm mũ và hàm Hyperbolic tangent (*ISMC_Tanh*) đã được thiết kế cho robot để khắc phục nhược điểm của điều khiển trượt cổ điển. Các thành phần phi tuyến trong luật điều khiển *ISMC_Tanh* được xấp xỉ bằng mạng nơ-ron RBF và tạo ra bộ điều khiển ISMC-RBF. Tính ổn định của hệ thống đã được chứng minh bằng lý thuyết Lyapunov. Các kết quả mô phỏng trong MATLAB/Simulink cho thấy đáp ứng thực tế x_w hội tụ về x_d , y_w hội tụ về y_d theo phương x và y tương ứng, hiện tượng chattering có biên độ nhỏ và tần số dao động thấp. Kết quả khảo sát với trường hợp ngõ vào tham chiếu thay đổi và có nhiễu tác động ở ngõ ra của robot bộ điều khiển ISMC-RBF đã cho thấy tính bền vững, sự phù hợp và độ tin cậy của bộ điều khiển này trong ứng dụng điều khiển bám quỹ đạo robot.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] 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.
- [2] 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, 2008.
- [3] T. H. Le, D. Tran, and V. T. Vu, "Adaptive Sliding Mode Control for Three-Wheel Omnidirectional Mobile Robot," *Int. J. Eng. Trends Technol.*, vol. 71, no. 5, pp. 9–17, 2023.
- [4] N. Hacene and B. Mendil, "Motion Analysis and Control of Three-Wheeled Omnidirectional Mobile Robot," *J. Control Autom. Electr. Syst.*, vol. 30, no. 2, pp. 194–213, 2019.
- [5] M. A. Kawtharani, V. Fakhari, and M. R. Haghjoo, "Tracking Control of an Omni-Directional Mobile Robot," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, Turkey: IEEE, 2020, pp. 1–8.
- [6] A. S. Andreev and O. A. Peregodova, "On Global Trajectory Tracking Control for an Omnidirectional Mobile Robot with a Displaced Center of Mass," *Nelineinaya Din.*, vol. 16, no. 1, pp. 115–131, 2020.
- [7] H. Mou, "Research On the Formation Method of Omnidirectional Mobile Robot Based On Dynamic Sliding Mode Control," *Acad. J. Manuf. Eng.*, vol. 18, pp. 148–154, 2020.
- [8] D. Nganga-Kouya, A. F. Okou, and J. M. L. N. Mezui, "Modeling and Nonlinear Adaptive Control for Omnidirectional Mobile Robot," *Res. J. Appl. Sci. Eng. Technol.*, vol. 18, no. 2, pp. 59–69, 2021.
- [9] 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.
- [10] 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, 2021.
- [11] C.-H. Lin and F.-Y. Hsiao, "Proportional-Integral Sliding Mode Control with an Application in the Balance Control of a Two-Wheel Vehicle System," *Applied Sciences*, vol. 10, no. 1, pp. 1–27, 2020.
- [12] J. Liu, *Sliding Mode Control Using MATLAB*. Academic Press, 2017.
- [13] 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 University Journal of Science and Technology*, vol. 12, no. 1, pp. 116–123, 2019.
- [14] J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation*. Springer Berlin, Heidelberg, 2013.
- [15] T. T. T. Tran, T. T. Pham, and C. N. Nguyen, "Performance evaluation of the reaching laws in sliding mode control for omnidirectional mobile robot," *the 6th Vietnam International Conference and Exhibition on Control and Automation –VCCA2021*, 2021, pp. 801–809.