RESEARCH OF TRAJECTORY TRACKING CONTROL FOR MOBILE ROBOT BASED ON REINFORCEMENT LEARNING TECHNIQUE

Roan Van Hoa[1*], Lai Khac Lai[2], Le Thi Thu Ha[2]
[1]*University of Economics – Technology for Industries*
[2]*TNU - University of Technology*

ARTICLE INFO

Received: 28/3/2022
Revised: 31/5/2022
Published: 31/5/2022

KEYWORDS

Reinforcement learning
Adaptive dynamic programming
Neural network
Hamilton Jacobi Bellman equation
Mobile robot

ABSTRACT

Currently, the use of mobile robots is increasingly popular in industries. One of the important problems in motion control of mobile robots is the control of tracking the reference motion trajectory. However, the mobile robot has a cascade control structure consisting of a dynamic controller in the inner ring and a kinematic controller in the outer ring. To solve the design problem without separating separate controllers, the paper presents a method using the online adaptive dynamic programming reinforcement learning technique with the structure using only a neural network approximating the function (OADP1NN). The algorithm can directly approximate the optimal solution (solution to the Hamilton Jacobi Bellman equation – HJB) simultaneously with the optimal control law. Performing simulations on Matlab software, the results showed that the OADP1NN algorithm has fully met two criteria for controlling robots: tracking the reference trajectory and minimizing the cost function related to tracking error and control energy.

NGHIÊN CỨU ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO ROBOT TỰ HÀNH TRÊN CƠ SỞ KỸ THUẬT HỌC TĂNG CƯỜNG

Roãn Văn Hóa[1*], Lại Khắc Lãi[2], Lê Thị Thu Hà[2]
[1]*Trường Đại học Kinh tế - Kỹ thuật Công nghiệp*
[2]*Trường Đại học Kỹ thuật Công nghiệp – ĐH Thái Nguyên*

THÔNG TIN BÀI BÁO

Ngày nhận bài: 28/3/2022
Ngày hoàn thiện: 31/5/2022
Ngày đăng: 31/5/2022

TỪ KHÓA

Học tăng cường
Quy hoạch động thích nghi
Mạng nơ ron
Phương trình HJB
Robot tự hành

TÓM TẮT

Hiện nay, việc sử dụng robot tự hành ngày càng phổ biến trong các ngành công nghiệp. Một trong những bài toán quan trọng về điều khiển chuyển động robot tự hành là điều khiển bám quỹ đạo chuyển động tham chiếu. Tuy nhiên, robot tự hành có cấu trúc điều khiển tầng bao gồm bộ điều khiển động lực học ở vòng trong và bộ điều khiển động học ở vòng ngoài. Để giải quyết bài toán thiết kế không cần chia tách bộ điều khiển riêng biệt, bài báo trình bày phương pháp sử dụng kỹ thuật học tăng cường quy hoạch động thích nghi trực tuyến với cấu trúc chỉ sử dụng một mạng nơ ron xấp xỉ hàm (Online adaptive dynamic programming with one neural network - OADP1NN). Thuật toán có thể xấp xỉ trực tuyến nghiệm tối ưu (nghiệm phương trình Hamilton Jacobi Bellman – HJB) đồng thời với luật điều khiển tối ưu. Thực hiện mô phỏng trên phần mềm Matlab, các kết quả cho thấy thuật toán OADP1NN đã đáp ứng đầy đủ được hai tiêu chí điều khiển robot tự hành đó là: bám quỹ đạo tham chiếu và tối thiểu hóa hàm chi phí liên quan đến sai số bám và năng lượng điều khiển.

DOI: https://doi.org/10.34238/tnu-jst.5759

[*] Corresponding author. *Email: rvhoa@uneti.edu.vn*

1. Giới thiệu

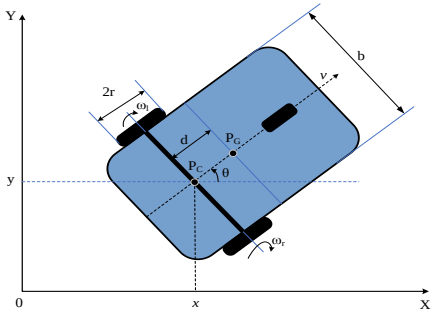
Trong những năm gần đây, điều khiển robot tự hành được quan tâm nghiên cứu và phát triển rộng rãi, đặc biệt là bài toán điều khiển bám quỹ đạo. Đã có nhiều phương pháp từ điều khiển kinh điển đến điều khiển hiện đại được đề xuất áp dụng cho robot tự hành. Các nghiên cứu trước đây thường sử dụng cấu trúc hai mạch vòng điều khiển: mạch vòng động học bên ngoài sử dụng hàm Lyapunov tổng hợp bộ điều khiển bám vị trí, mạch vòng động lực học bên trong điều khiển bám tốc độ. Trong tài liệu [1], kỹ thuật cuốn chiếu được sử dụng, tuy nhiên tham số bộ điều khiển động học được chọn qua thực nghiệm hoặc bằng kinh nghiệm của người thiết kế sao cho cân bằng được cả hai tiêu chí về chất lượng bám lẫn năng lượng điều khiển từ mô men ở bánh xe. Cách chọn tham số như vậy sẽ không tối thiểu hóa được hàm chỉ tiêu chất lượng liên quan đến chất lượng bám quỹ đạo và năng lượng điều khiển. Phương pháp tuyến tính hóa hồi tiếp thích nghi được đề xuất [2], trong đó việc chọn tham số cho luật điều khiển động học được lược bỏ và không cần đến hai bộ điều khiển động học và động lực học riêng biệt. Tuy nhiên, phương pháp này không giải quyết được bài toán tối ưu. Đặc trưng quan trọng của robot tự hành là mô hình có thể biểu diễn được ở dạng hệ thống phi tuyến hồi tiếp chặt (strictly feedback form [3]). Hệ thống này đã được khai thác để thiết kế luật điều khiển tối ưu cho robot tự hành mà không cần chia tách bộ điều khiển. Về mặt toán học, bài toán điều khiển tối ưu được giải nếu nghiệm phương trình Hamilton-Jacobi-Bellman (HJB) được giải. Đối với hệ phi tuyến, phương trình HJB nhìn chung không thể giải được [4]. Từ đó, nhiều giải thuật xấp xỉ nghiệm HJB online dựa trên lý thuyết cơ sở của học tăng cường (Reinforcement learning – RL) đã được đề xuất. Các nghiên cứu [5]-[7] xấp xỉ thích nghi online nghiệm HJB cho hệ phi tuyến. Các phương pháp này sử dụng giải thuật PI (Policy Iteration) với cấu trúc ADP chuẩn gồm hai xấp xỉ hàm, đó là hai mạng nơ ron truyền thẳng trong [8]-[10]. Luật cập nhật trọng số mạng nơ ron trong các phương pháp này được chứng minh ổn định (Uniform Ultimate Bounded - UUB [11]), trong quá trình xấp xỉ online cùng với hàm chi phí và luật điều khiển hội tụ về giá trị cận tối ưu. Tuy nhiên, sử dụng hai xấp xỉ hàm thì tốc độ hội tụ, chi phí tính toán và tài nguyên lưu trữ vẫn còn là vấn đề thách thức.

Để khắc phục hạn chế sử dụng nhiều xấp xỉ hàm trong cấu trúc điều khiển, bài báo sử dụng thuật toán OADP1NN với cấu trúc điều khiển chỉ sử dụng duy nhất một xấp xỉ hàm. Trong đó, luật cập nhật tham số và thuật toán điều khiển sẽ được thiết kế phù hợp để cải thiện được tốc độ hội tụ, bên cạnh đó nghiệm HJB vẫn được xấp xỉ và hệ kín luôn duy trì ổn định.

Phần tiếp theo của bài báo được trình bày như sau: phần hai là mô hình toán học của robot tự hành đã được biểu diễn ở dạng hệ thống phi tuyến hồi tiếp chặt. Phần ba thuật toán học tăng cường OADP1NN được giới thiệu để điều khiển bám quỹ đạo cho robot tự hành. Kết quả mô phỏng được đưa ra trong phần bốn. Cuối cùng, phần thứ năm là kết luận của bài báo này.

2. Mô hình toán học của robot tự hành

Robot tự hành ba bánh (hai bánh chủ động, một bánh lái) có thể chuyển động thẳng và quay trên mặt phẳng dựa vào mô men xoắn từ hai cơ cấu chấp hành độc lập bố trí tại bánh xe robot [2]. Mô hình robot tự hành ba bánh và hệ trục tọa độ được mô tả trên hình 1.



Hình 1. *Mô hình robot tự hành ba bánh và hệ trục tọa độ*

Khối lượng của robot tập trung tại trọng tâm C bao gồm khối lượng khung không kể các bánh xe và khối lượng các bánh xe qui đổi. Bề rộng của robot là b, bán kính của mỗi bánh xe là r. Khoảng cách giữa tâm và trục dẫn động là d. Tọa độ trọng tâm robot so với hệ qui chiếu OXY cố định trên mặt phẳng, hướng di chuyển, véc tơ vận tốc tịnh tiến và vận tốc quay lần lượt được kí hiệu là x, y, θ, v, ω.

Robot tự hành tổng quát trong không gian cấu hình n chiều với tọa độ suy rộng $q = [q_1, q_2, .., q_n] \in \mathbb{R}^n$, chịu m ràng buộc với $m < n$ được biểu diễn dưới dạng $A(q)\dot{q} = 0$ với $A(q) \in \mathbb{R}^{mxn}$ là ma trận đủ hạng [12]. Giả sử rằng $S(q) \in \mathbb{R}^{n\times(n-m)}$ cũng là ma trận đủ hạng được tạo thành từ trường véc tơ trơn và độc lập tuyến tính trong không gian rỗng của $A(q)$ sao cho $A(q).S(q) = 0$.

Gọi $\Theta(t) = [v^T, \omega^T] \in \mathbb{R}^{n-m}$ véc tơ vận tốc, phương trình chuyển động của robot tự hành dựa vào hai ràng buộc của $A(q)$ có thể viết thành:

$$\dot{q} = S(q)\Theta(t) \tag{1}$$

Để có phương trình động học robot, ta sử dụng phương trình Lagrange [12]:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial \dot{q}}\right) = F_T \tag{2}$$

Trong đó: F_T là véc tơ lực suy rộng, L là hàm Lagrange. Giả sử, robot tự hành chuyển động trên nền phẳng nên L chỉ chứa động năng:

$$L = \sum_{i=1}^{l_k} v_i^T m_i v_i + \omega_i^T I_i \omega_i \tag{3}$$

Trong đó: l_k số khâu trong robot tự hành, v_i, ω_i, m_i, I_i lần lượt là véc tơ vận tốc tịnh tiến, vận tốc quay, khối lượng và mô men quán tính của khâu thứ i. Từ đó, phương trình động học của robot tự hành trở thành:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B(q)F(q) + B(q)\tau_m = B(q)\tau - A^T(q)\lambda \tag{4}$$

Trong đó: $M(q) \in R^{n\times n}$ là ma trận đối xứng xác định dương, $C(q,\dot{q}) \in R^{n\times n}$ là ma trận lực Coriolis và ly tâm, $F(\dot{q}) \in R^{n-m}$ là véc tơ lực ma sát, $\tau_m \in R^{n-m}$ là nhiễu mô men từ cơ cấu chấp hành. $B(q) \in R^{n\times(n-m)}$ là ma trận chuyển đổi, $\tau \in R^{n-m}$ là véc tơ mô men điều khiển, $\lambda \in R^{1\times m}$ là véc tơ lực ràng buộc. Đạo hàm phương trình (1) ta có:

$$\ddot{q} = \dot{S}(q)\,\Theta + S(q)\dot{\Theta} \tag{5}$$

Lưu ý rằng: $A(q)S(q) = 0$

$$\bar{M}(q)\dot{\Theta}(t) + \bar{C}(q,\dot{q})\Theta(t) + \bar{F}(\dot{q}) + \bar{\tau}_m = \bar{B}(q)\tau \tag{6}$$

Trong đó:

$$\bar{M}(q) = S^T M S, \bar{C}(q,\dot{q}) = S^T M S + S^T C S, \bar{B}(q) = S^T B(q), \bar{F}(\dot{q}) = S^T M S\dot{\Theta} + \bar{B}(q)F, \bar{\tau}_m = B(q)\tau_m \tag{7}$$

Theo [3], một số tính chất cần thiết của các thành phần trong mô hình toán được trình bày nhằm mục đích xác định tính tương thích khi áp dụng giải thuật OADP1NN.

Tính chất 1: $\bar{M}(q)$ là ma trận đối xứng xác định dương bị chặn thỏa mãn điều kiện $\bar{m}_{min} \leq \|\bar{M}(q)\| \leq \bar{m}_{max}$ với $\bar{m}_{min}$ và $\bar{m}_{max}$ là các hằng số dương.

Tính chất 2: $\bar{C}(q,\dot{q})$ bị chặn thỏa mãn điều kiện $\|\bar{C}(q,\dot{q})\| \leq c_{max}$, với c_{min} , c_{max} là các hằng số dương.

Tính chất 3: $\bar{F}(\dot{q})$ bị chặn thỏa mãn điều kiện $\bar{F}(\dot{q}) \leq f_{max}\|\dot{q}\|$, với f_{max} là hằng số dương.

Tính chất 4: Nhiễu mô men τ_m có năng lượng hữu hạn, nghĩa là $\bar{\tau}_m \in L_2[0,T], 0 < T < \infty$, τ_m bị chặn sao cho$\|\bar{\tau}_m\| \leq \bar{\tau}_{mmax}$ với $\bar{\tau}_m$ là hằng số dương.

Đặt các hàm:

$F_q(q) = 0_{n\times 1}$, $G_q(q) = S(q) \in \mathbb{R}^{n\times(n-m)}$, $F_\Theta(q,\Theta) = -\bar{M}^{-1}(q)\,(\bar{C}(q,\dot{q})\Theta + \bar{F}(\dot{q})) \in \mathbb{R}^{(n-m)}$, $k_\Theta(q,\Theta) = \bar{M}^{-1}(q)\ \mathbb{R}^{(n-m)\times(n-m)}$, $G_\Theta(q,\Theta) = \bar{M}^{-1}(q)\bar{B}(q) \in \mathbb{R}^{(n-m)\times(n-m)}$.

Trong bài toán đang xét, ta bỏ qua nhiễu τ_m

Ta có phương trình không gian trạng thái của robot tự hành dưới dạng hệ phi tuyến hồi tiếp chặt như sau:

$$\begin{cases} \dot{q} = F_q(q) + G_q(q)\Theta \\ \dot{\Theta} = F_\Theta(q,\Theta) + G_\Theta(q,\Theta)\tau \end{cases} \quad (8)$$

Kết hợp với tính chất từ 1 đến 4, ta có một số tính chất cần thiết về các thành phần động học trong mô hình:

Tính chất 5: $F_\Theta(q,\Theta) \le \bar{m}_{min}^{-1}(c_{max} + \bar{f}_{1max}\, s_{max})\|\Theta\|$, trong đó s_{max} là chặn trên của $\|S(q\|)$.

Tính chất 6: $g_q(q)$ là ma trận bị chặn thỏa mãn điều kiện $g_{min} \le \|G_q(q)\| \le g_{max}$ với g_{min} và g_{max} là các hằng số dương.

Tính chất 7: $\bar{B}(q)$ là ma trận không suy biến chứa tham số hằng, đó là bán kính r của các bánh xe và độ rộng khung robot b.

Tính chất 8: $G_\Theta(q,\Theta)$ bị chặn thỏa điều kiện $\bar{m}_{max}^{-1}\bar{B} \le \|G_\Theta(q,\Theta)\| \le \bar{m}_{min}^{-1}\bar{B}$. Kết hợp với tính chất 1, ta có $G_\Theta(q,\Theta) \ne 0$.

Tính chất 9: $F_\Theta(q,\Theta)$, $G_q(q)$, $G_\Theta(q,\Theta)$ là các hàm phi tuyến trơn.

Nếu cho trước robot tham chiếu có mô hình như sau:

$$\dot{q}_d = G_q(q_d)\,\Theta_{rd} \quad (9)$$

Trong đó, $\dot{q}_d = [x_d, y_d, \Theta_d]^T$ là quỹ đạo trơn, bị chặn thỏa mãn ràng buộc $\dot{q}_d = g_q(q_d)\,\Theta_{rd} = S(q_d)\,\Theta_{rd}$ với Θrd là véc tơ vận tốc giả sử khả vi liên tục biết trước. Mục tiêu của bài toán là thiết kế luật điều khiển để quỹ đạo hệ thống phương trình (8) bám quỹ đạo phương trình (9) đồng thời thỏa mãn hai yêu cầu:

- Tích hợp chung luật điều khiển động học và động lực học.
- Tối thiểu hàm chi phí liên quan đến sai số bám bị ràng buộc bởi hệ thống.

Các bước biến đổi sau đây được thực hiện để có phương trình động lực học bám thích hợp nhằm mục đích thiết kế luật điều khiển [3].

$$\dot{q} - \dot{q}_d = \dot{e}_q = -\dot{q}_d + G(q)(\Theta - \Theta_d) + G(q)\Theta_d = f_{eq}(t) + G_q(q)\Theta_d^* + G_q(q)e_\Theta \quad (10)$$

Trong đó, $f_{eq}(t) = 0_{n\times 1}$, $e_\Theta = \theta - \theta_d \in \mathbb{R}^{(n-m)}$ với $\Theta_d \in \mathbb{R}^{(n-m)}$là ngõ vào điều khiển ảo sao cho $\Theta_d = \Theta_d^* + \Theta_{da}$ với $\Theta_d^* \in \mathbb{R}^{(n-m)}$ là véc tơ tín hiệu điều khiển bám tối ưu và Θ_{da} là nghiệm của phương trình:

$$G_q(q)\,\Theta_{da} - G_q(q_d)\,\Theta_{rd} = 0 \quad (11)$$

Tương tự:

$$\begin{aligned} \dot{\Theta} - \dot{\Theta}_d = \dot{e}_\Theta &= -\dot{\Theta}_d + F_\Theta(q,\Theta) + G_\Theta(q,\Theta)\tau \\ &= f_{e\Theta}(t) + G_\Theta(q,\Theta)\,\tau^* - g_q^T(q)\,e_q \end{aligned} \quad (12)$$

Trong đó, Θ_d là véc tơ vận tốc mong muốn, xác định dựa vào mô hình chuẩn từ phương trình thứ hai trong (8), đó là $\Theta_d = F_\Theta(q_d,\Theta_d) + G_\Theta(q_d,\Theta_d)\,\tau_d$, $f_{e\Theta}(t) = F_\Theta(q,\Theta) - F_\Theta(q_d,\Theta_d)$, τ^* là véc tơ tín hiệu mô men điều khiển tối ưu được thiết kế sao cho $\tau = \tau^* + \tau_d$ với τ_d là nghiệm của phương trình:

$$(G_\Theta(q,\Theta) - G_\Theta(q_d,\Theta_d)\,\tau_d + g_q^T(q)\,e_q = 0 \quad (13)$$

Đặt $x = [q^T, \Theta^T]^T \in R^{2n-m}$, $e = \left[e_q{}^T, e_\Theta{}^T\right]^T \in R^{2n-m}$, $f_e(t) = \left[f_{eq}{}^T, f_{e\Theta}{}^T\right]^T \in R^{2n-m}$, $u^* = u - u_a$, trong đó $u^* = \left[\Theta_d^{*T}, \tau^{*T}\right]^T \in R^{2(n-m)}$, $u = \left[\Theta_d^T, \tau^T\right]^T \in R^{2(n-m)}$ và $u_a = \left[\Theta_{da}^T, \tau_d^T\right]^T \in R^{2(n-m)}$, $G(x) = diag\left[G_q(q), G_\Theta(q,\Theta)\right] \in R^{(2n-m)\times 2(n-m)}$

Với cách đặt như trên ta đi tiến hành thiết kế bộ điều khiển tối ưu thích nghi trực tuyến cho hệ như sau:

$$\dot{e} = F(e) + G(x)u^* \quad (14)$$

3. Thuật toán học tăng cường OADP1NN

Lý thuyết học tăng cường RL kế thừa từ lý thuyết tối ưu của qui hoạch động (Dynamic Programming - DP) và phát triển thành lý thuyết qui hoạch động thích nghi (Adaptive Dynamic Programming - ADP) hoặc qui hoạch động xấp xỉ (Approximate Dynamic Programming - ADP). ADP đã khắc phục được các hạn chế của DP như điều khiển off-line, không điều khiển thời gian thực, cần mô hình toán chính xác. Ngoài ra, khi ADP sử dụng xấp xỉ hàm sẽ khắc phục được các điểm yếu quan trọng của DP như giảm chi phí tính toán và tài nguyên lưu trữ, khắc phục được hiện tượng bùng nổ tổ hợp (Curse of Dimensionality - COD) khi rời rạc hóa không gian trạng thái, đặc biệt nếu đối tượng điều khiển là hệ MIMO (Multi Input Multi Output).

3.1. Cấu trúc điều khiển và luật cập nhật tham số online

Thuật toán OADP1NN sử dụng để xấp xỉ online nghiệm $V^*(e)$ (nghiệm HJB) đồng thời với luật điều khiển tối ưu $u^*(e)$. Cấu trúc điều khiển OADP1NN được phát triển dựa trên cấu trúc cơ sở ADP sử dụng hai mạng nơ ron [8]. Tuy nhiên, điểm khác biệt quan trọng là OADP1NN không sử dụng mạng nơ ron cho luật điều khiển. Mạng nơ ron được sử dụng để xấp xỉ hàm đánh giá $V(e)$ và được định nghĩa:

$$V(e) = W^T\varphi(e) + \varepsilon(e) \tag{15}$$

Trong đó, W là trọng số mạng nơ ron, $\varphi(e)$: $\mathbb{R}^n \to \mathbb{R}^{n_h}$ là véc tơ hàm tác động, với n_h là số đơn vị nút ở lớp ẩn và $\varepsilon(e)$ là sai số xấp xỉ của mạng neural. Với mạng nơ ron truyền thẳng một lớp, ta có thể chọn $\varphi(e)$ sao cho $n_h \to \infty, \varepsilon \to 0$ và $\varepsilon_e = \partial\varepsilon/\partial e \to 0$, ngoài ra với n_h hữu hạn thì $\|\varepsilon(e)\| < \varepsilon_{max}$và $\|\varepsilon_e\| < \varepsilon_{emax}$, với ε_{max} và ε_{emax} là các hằng số dương

Định nghĩa hàm Hamilton:

$$H(e,u,V_e) = V_e^T(F(e) + G(x)u) + Q(e) + u^TRu \tag{16}$$

Hàm Hamilton trở thành:

$$H(e,u,W) = W^T\varphi_e(F(e) + G(x)u) + Q(e) + u^TRu - \varepsilon_H = 0 \tag{17}$$

Trong đó, $\varphi_e = \partial\varphi/\partial e \in \mathbb{R}^{n_h xn}$ và ε_H được xác định:

$$\varepsilon_H = -\varepsilon_e[F(e) + G(x)u] \tag{18}$$

Phương trình HJB theo tham số V_e^* [8]:

$$\begin{cases} Q(e) + (V_e^*)^TF(e) - \dfrac{1}{4}(V_e^*)^TG(x)R^{-1}G^T(x)V_e^* = 0 \\ V^*(0) = 0 \end{cases} \tag{19}$$

Sử dụng mạng nơ ron (15) cho phương trình HJB ta có:

$$Q(e) + W^T\varphi_eF(e) - \frac{1}{4}W^T\varphi_eG\varphi_e^TW + \varepsilon_{HJB} = 0 \tag{20}$$

Trong đó, ε_{HJB} là sai số thặng dư gây bởi mạng nơ ron:

$$\varepsilon_{HJB} = \varepsilon_e^TF(e) - \frac{1}{2}W^T\varphi_eG\varepsilon_e - \frac{1}{4}\varepsilon_e^TG\varepsilon_e \tag{21}$$

Sử dụng luật điều khiển tối ưu:

$$u^* = -\frac{1}{2}R^{-1}G^T(x)V_e^* \tag{22}$$

Với $\boldsymbol{G}(x) = G(x)R^{-1}G^T(x) \in \mathbb{R}^{nxn}$ và $\boldsymbol{G}(x) = G^T(x) > 0$. Cộng và trừ (21) với $\frac{1}{2}\varepsilon_e^TG\varepsilon_e$, sử dụng luật điều khiển tối ưu (22) và để ý đạo hàm của (15), ta có:

$$\varepsilon_{HJB} = \varepsilon_e^T(F(e) + Gu^*) + \frac{1}{4}\varepsilon_e^TG\varepsilon_e \tag{23}$$

Tính chất 10:

$$G_{min} \le \|\boldsymbol{G}(x)\| \le G_{max} \tag{24}$$

Trong đó, $G_{min} = \lambda_{max}(R)G_{min}^2, G_{max} = \lambda_{min}(R)G_{max}^2$, với λ_{min} và λ_{max} lần lượt là giá trị riêng lớn nhất và nhỏ nhất của ma trận R.

Tính chất 11: Khi $n_h \to \infty$, ε_{HJB} hội tụ đều về giá trị không, với n_h hữu hạn, ε_{HJB} bị chặn trong tập đóng.

Trọng số lý tưởng mạng nơ ron (15) chưa xác định, do đó $V(e)$ sẽ được xấp xỉ bởi hàm số sau:

$$\hat{V}(e) = \widehat{W}^T \varphi(e) \tag{25}$$

Trong đó $\widehat{W} \in \mathbb{R}^{n_h}$ là trọng số của mạng nơ ron xấp xỉ.

Sử dụng $\hat{V}(e)$ cho phương trình mục tiêu:

$$V_e^T(F(e) + G(x)u) + Q(e) + u^T R u = 0, V(0) = 0 \tag{26}$$

Gọi $e_1$là sai số của Hamilton (16) gây bởi mạng nơ ron xấp xỉ, ta có:

$$H\left(e, u, \widehat{W}\right) = \widehat{W}^T \varphi_e(F(e) + G(x)u) + Q(e) + u^T R u - \varepsilon_H = e_1 \tag{27}$$

Sai số xấp xỉ trọng số: $\widetilde{W} = W - \widehat{W}$. Từ (17) và (27) ta có:

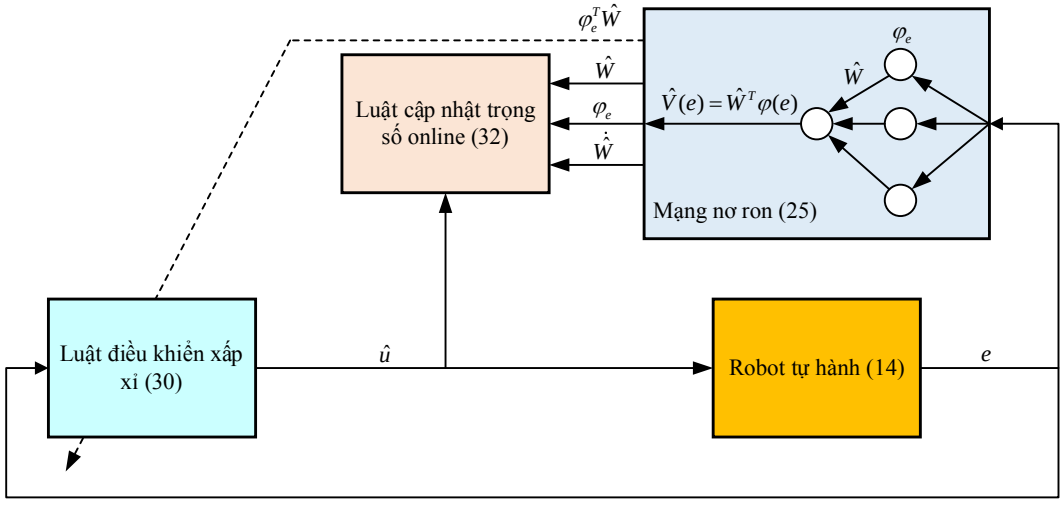
$$e_1 = -\widetilde{W}^T \varphi_e(F(e) + G(x)u) + \varepsilon_H \tag{28}$$

Với bất kỳ luật điều khiển $u \in U(e)$ cho trước, để $\widehat{W} \to$ W, khi đó $e_1 \to \varepsilon_H$, ta cần chỉnh định $\widehat{W}$ nhằm tối thiểu $E_1 = \frac{1}{2} e_1^T e_1$. Sử dụng giải thuật Normalized gradient descent, luật cập nhật được $\widehat{W}$ định nghĩa như sau:

$$\dot{\widehat{W}} = -\alpha_1 \frac{\partial E_1}{\partial \widehat{W}} = -\alpha_1 \frac{\sigma}{(\sigma^T \sigma + 1)^2} \left(\sigma^T \widehat{W} + Q(e) + u^T R u\right) \tag{29}$$

Trong đó: $\sigma = \varphi_e(F(e) + G(x)u)$. Mẫu số bình phương của (29) nhận được bởi giải thuật Levenberg–Marquardt cải tiến bằng cách thay $\sigma^T \sigma + 1$ bằng$(\sigma^T \sigma + 1)^2$.

Trong thuật toán AC (Actor – Critic) [8], xấp xỉ hàm sử dụng luật cập nhật (29), trong đó u được thay bởi xấp xỉ hàm bộ điều khiển. Vì vậy, cần hai luật cập nhật khác nhau. Luật cập nhật xấp xỉ hàm nhằm tối thiểu sai số bình phương xấp xỉ hàm trong khi luật cập nhật của bộ điều khiển bảo đảm sự ổn định của toàn hệ kín. Ngược lại, giải thuật OADP1NN trong bài báo chỉ sử dụng duy nhất một mạng nơ ron nên luật cập nhật (22) không thể áp dụng trực tiếp, cần đề xuất mới theo hướng kết hợp cả hai mục tiêu trên vào một luật cập nhật trọng số xấp xỉ hàm duy nhất. Cấu trúc điều khiển trong thuật toán OADP1NN được mô tả trên hình 2.



Hình 2. *Cấu trúc điều khiển OADP1NN*

Với hàm đánh giá xấp xỉ $\hat{V}(e)$ (25) luật điều khiển xấp xỉ sẽ là:

$$\hat{u} = -\frac{1}{2} R^{-1} G^T(x) \hat{V}_e = -\frac{1}{2} R^{-1} G^T(x) \varphi_e^T \widehat{W} \tag{30}$$

Sử dụng (25) và (30) cho phương trình mục tiêu (26), gọi e_2 sai số của hàm Hamilton (16) sinh ra bởi mạng nơ ron xấp xỉ và luật điều khiển xấp xỉ, ta có:

$$H\left(e, \hat{u}, \widehat{W}\right) = \widehat{W}^T \varphi_e F(e) + Q(e) + \hat{u}^T R \hat{u} = e_2 \tag{31}$$

Luật cập nhật $\widehat{W}$ nhằm tối thiểu sai số $E_2 = \frac{1}{2}e_2^T e_2$, ổn định hệ kín được đề xuất:

$$\dot{\widehat{W}} = \begin{cases} \dot{\widehat{W}}_1 & nếu\ x^T(F(e) + G(x)\hat{u}) \le 0 \\ \dot{\widehat{W}}_1 + W_{RB} & ngược\ lại \end{cases} \quad (32)$$

Trong đó:

$$\dot{\widehat{W}}_1 = -\alpha_1 \frac{\partial E_2}{\partial \widehat{W}} = -\alpha_1 \frac{\hat{\sigma}}{(\hat{\sigma}^T\hat{\sigma} + 1)^2}\left(\hat{\sigma}^T\widehat{W} + Q(e) + \hat{u}^T R\hat{u}\right) \quad (33)$$

$$W_{RB} = \frac{1}{2}\alpha_2 \varphi_e \boldsymbol{G}(x)e \quad (34)$$

Với $\sigma = \varphi_e(F(e) + G(x)\hat{u})$. Luật cập nhật $\widehat{W}_1$ được thiết kế dựa vào giải thuật Levenberg-Marquardt cải tiến sử dụng $(\hat{\sigma}^T\hat{\sigma} + 1)^2$ thay cho $(\hat{\sigma}^T\hat{\sigma} + 1)$. Luật bền vững W_{RB} được thêm vào nhằm đảm bảo hệ thống ổn định theo tiêu chuẩn bị chặn UUB.

3.2. Phân tích ổn định và hội tụ của giải thuật OADP1NN

Giả định 1: Động học hệ thống $F(e)$ giả sử thỏa điều kiện Lipschitz, sao cho $\|F(e)\| \le a\|e\|$.

Giả định 2: Phương trình (14) với luật điều khiển u^* bị chặn bởi hằng số dương μ: $\|F(e) + G(x)u^*\| \le \mu$

Sự hội tụ tham số và tính ổn định của hệ kín trong giải thuật OADP1NN thông qua Định lý 1:

Định lý 1: Xét hệ thống phi tuyến (14), phương trình HJB (19), mạng nơ ron để xấp xỉ hàm đánh giá (25), luật điều khiển (30) và luật cập nhật trọng số mạng nơ ron (32), thì thuật toán OADP1NN bảo đảm [8]:

- Ổn định: Toàn bộ trạng thái của hệ kín (14) và sai số xấp xỉ mạng nơ ron trong giải thuật OADP1NN sẽ bị chặn UUB.

- Hội tụ: Khi t → ∞, sai số giữa hàm chi phí xấp xỉ so với tối ưu thỏa tiêu chuẩn $\left\|\hat{V} - V*\right\| \le \varepsilon_V$, với ε_V là hằng số dương nhỏ, và sai số giữa luật điều khiển xấp xỉ so với tối ưu thỏa tiêu chuẩn $\|\hat{u} - u*\| \le \varepsilon_u$, với ε_u là hằng số dương nhỏ.

4. Kết quả mô phỏng

Để kiểm chứng tính hiệu quả của thuật toán điều khiển, ta tiến hành mô phỏng trên phần mềm Matlab trong hai trường hợp. Với các tham số của robot tự hành được chọn như sau [2]:

I = 5 kg.m^2, b = 0,5 m, r = 0,05 m, d = 0,15, $m = 10\ kg$, $K_2 = 0{,}267$

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}, \bar{M} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \bar{C} = \begin{bmatrix} \frac{2K_2}{r^2} & md\dot{\theta} \\ -md\dot{\theta} & \frac{2b^2K_2}{r^2} \end{bmatrix}, \bar{B} = \begin{bmatrix} \frac{1}{r} & \frac{b}{r} \\ \frac{1}{r} & -\frac{b}{r} \end{bmatrix}$$

Trạng thái vị trí ban đầu của WRM:

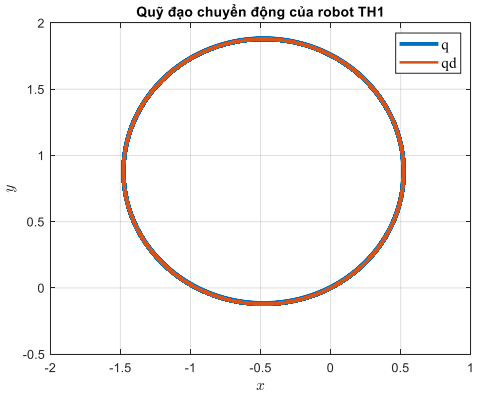
$q(0) = [0.5,0.5,0]^T, v(0) = [0,0]^T$

$$\varphi(e) = \begin{bmatrix} e_x^2, e_xe_y, e_xe_\theta, e_xe_v, e_xe_w, e_y^2, e_ye_\theta, \\ e_ye_v, e_ye_w, e_\theta^2, e_\theta e_v, e_\theta e_w, e_v^2, e_ve_w, e_w^2 \end{bmatrix}^T, \alpha_1 = 1, \alpha_2 = 0.01, \widehat{W} = zeros(15,1)$$

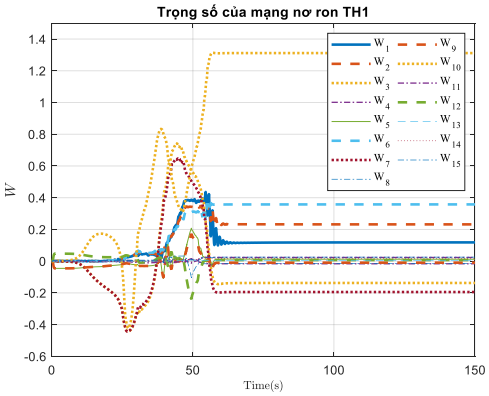
Trường hợp 1: Quỹ đạo hình tròn

Quỹ đạo mẫu :

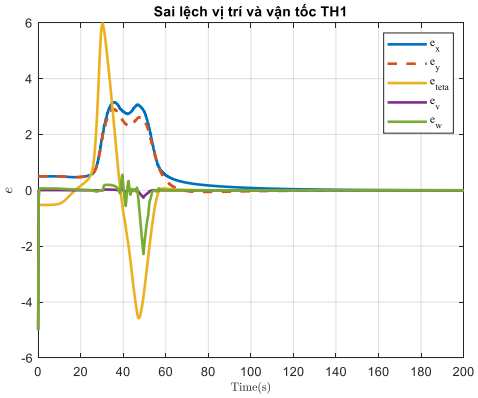
$\Theta_{rd} = \begin{pmatrix} 5 \\ 5 \end{pmatrix}$, $q_d = [x_d, y_d, \theta_d]^T$ được tạo với Θ_{rd} bên trên và bằng ràng buộc theo (9). Ta có kết quả mô phỏng bằng phần mềm Matlab như sau:



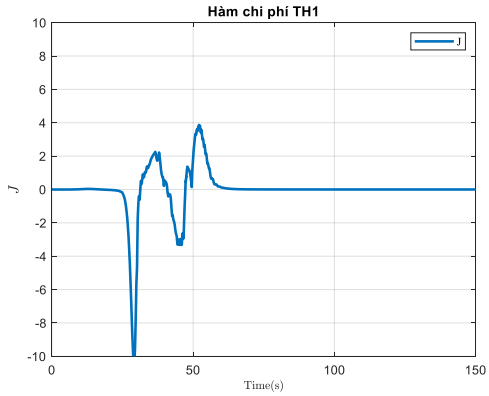
Hình 3. *Quỹ đạo đặt và quỹ đạo thực tế của robot TH1*



Hình 4. *Trọng số của mạng nơ ron TH1*



Hình 5. *Sai lệch vị trí của robot TH1*

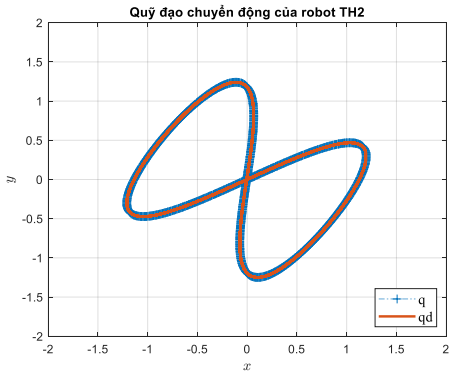


Hình 6. *Hàm chi phí TH1*

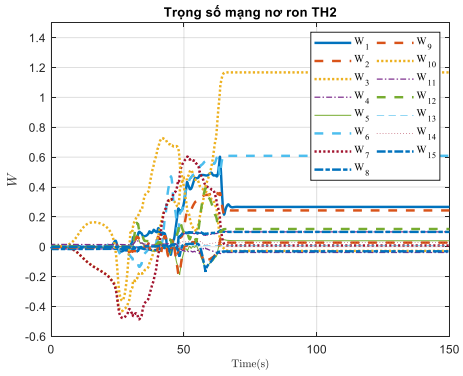
Trường hợp hai: Quỹ đạo hình số 8

Quỹ đạo mẫu:

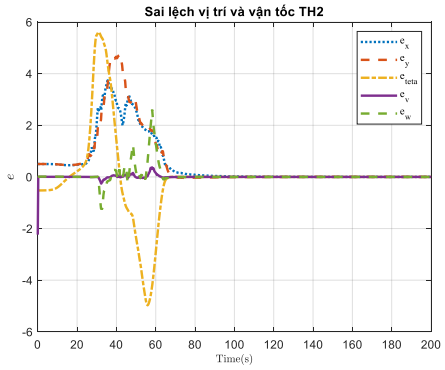
$$\Theta_{rd} = \begin{pmatrix} \sqrt{cos^2 t + 4cos^2(2t)} \\ (2\sin(t)\cos(2t) - 4\sin(2t)\cos(t))/(cos^2 t + 4cos^2(2t)) \end{pmatrix}$$



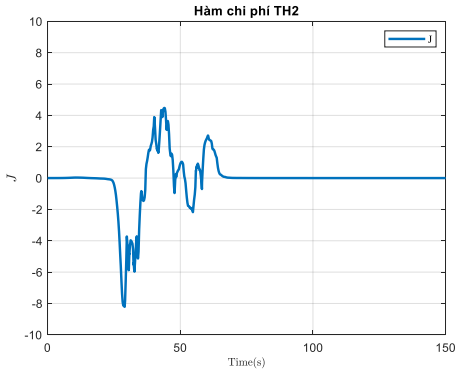
Hình 7. *Quỹ đạo đặt và quỹ đạo thực tế của robot TH2*



Hình 8. *Trọng số của mạng nơ ron TH2*



Hình 9. *Sai lệch vị trí của robot TH2*



Hình 10. *Hàm chi phí TH2*

Từ kết quả mô phỏng hình 3, hình 7 với hai quỹ đạo mẫu khác nhau, tuy nhiên ta thấy ở cả hai trường hợp bộ điều khiển đã cho chất lượng bám rất tốt. Bên cạnh đó hình 4, hình 8 thể hiện trọng số W của mạng nơ ron đã hội tụ trong quá trình học (65s đầu tiên) về một giá trị xác lập. Sai lệch giữa giá trị đặt và trạng thái thực tế của robot được thể hiện trong hình 5, hình 9 có thể thấy rằng bộ điều khiển OADP1NN đã bám quỹ đạo tham chiếu yêu cầu. Không chỉ đáp ứng được tiêu chí bám quỹ đạo tham chiếu mà bộ điều khiển còn tối thiểu hóa hàm chi phí liên quan đến sai số bám và năng lượng điều khiển thể hiện trong hình 6, hình 10. Như vậy, thuật toán đã giải quyết được bài toán điều khiển bám quỹ đạo tối ưu cho robot tự hành rất phù hợp để đưa vào ứng dụng điều khiển robot trong dân dụng và công nghiệp.

5. Kết luận

Thông qua việc phân tích, thiết kế và mô phỏng bộ điều khiển trên phần mềm Matlab có thể thấy được tính hiệu quả của thuật toán OADP1NN trong việc giải quyết đồng thời bài toán tối ưu và thích nghi trong điều khiển. Cấu trúc điều khiển được đề xuất chỉ sử dụng một mạng nơ ron nên làm giảm khối lượng tính toán so với cấu trúc sử dụng hai mạng nơ ron. Hơn nữa, cấu trúc này tránh được độ phức tạp của mô hình và tăng tốc độ hội tụ của thuật toán do các tham số mạng nơ ron và bộ điều khiển được cập nhật đồng thời. Ứng dụng của bộ điều khiển cho thấy khả năng giải quyết được, không chỉ bài toán tối ưu thông thường, mà còn xử lý được bài toán bám quỹ đạo có tính tới yếu tố tối ưu, một bài toán không phải đơn giản để giải quyết. Cuối cùng, định hướng phát triển về thực nghiệm là sử dụng phương pháp này để điều khiển robot tự hành ứng dụng trong dân dụng và công nghiệp.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] H. Hoang, "Direct adaptive control for trajectory tracking of mobile robot," *Proceeding of International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2012, pp. 300-305.

[2] S. Khoshnam and M. a. A. T. Alireza, "Adaptive feedback linearizing control of nonholonomic wheeled mobile robots in presence of parametric and nonparametric uncertainties," *Robotics and Computer-Integrated Manufacturing*, vol. 27, pp. 194-204, 2011.

[3] H. Zargarzadeh, T. Dierks, and S. Jagannathan, "Adaptive neural network based optimal control of nonlinear continuous-time systems in strict feedback form," *International Journal of Adaptive Control and Signal Processing*, vol. 28, pp. 305-324, 2014.

[4] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32-50, 2009.

[5] T. Dierks and S. Jagannathan, "Neural network output feedback control of robot formations," *IEEE Trans, Systems, Man, and Cybernetics*, vol. 40, pp. 383-399, 2010.

[6] T. Dierks and S. Jagannathan, "Optimal control of affine nonlinear continuous-time systems using an online Hamilton-Jacobi-Isaacs formulation," *Proceedings of 49th IEEE Conference on Decision and Control*, 2010, pp. 3048-3053.

[7] T. Dierks, B. Brenner, and S. Jagannathan, "Neural Network-Based Optimal Control of Mobile Robot Formations With Reduced Information Exchange," *IEEE Transactions on Control Systems Technology*, vol. 21, pp. 1407-1415, 2013.

[8] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, no. 46, pp. 878-888, 2010.

[9] Y. Jiang and Z. Jiang, "Robust adaptive dynamic programming and feedback stabilization of Nonlinear Systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, pp. 882-893, 2014.

[10] F. a. V. D. Lewis, "Reinforcement learning and adaptive dynamic," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32-50, 2009.

[11] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural network control of robot manipulators and nonlinear systems*, Taylor & Francis, 1999.

[12] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 9, pp. 589-600, 1998.