

ĐIỀU KHIỂN TRƯỢT NƠ RON THÍCH NGHI BỀN VỮNG CHO ROBOT BA BẬC TỰ DO

Thái Hữu Nguyên^{1,*}, Phan Xuân Minh², Nguyễn Công Khoa²

¹Trường Đại học Sư phạm Kỹ thuật Vinh,

²Bộ môn Điều khiển, Viện Điện, Trường Đại học Bách khoa Hà Nội

*Email: thainguyenktv@yahoo.com

Đến Tòa soạn: 26/12/2013; Chấp nhận đăng: 18/7/2014

TÓM TẮT

Bài báo trình bày một thuật toán thiết kế bộ điều khiển thích nghi bền vững cho đối tượng truyền ngược chặt bất định hàm số. Luật điều khiển được tổng hợp dựa trên nguyên lý điều khiển trượt. Đặc tính bất định của đối tượng được xấp xỉ bằng mạng nơ ron hướng tâm hai lớp, các trọng số của mạng được huấn luyện on-line. Bộ điều khiển được áp dụng cho mô hình Robot ba bậc tự do (DOF). Các kết quả mô phỏng cho thấy ưu điểm và khả năng ứng dụng trong thực tế của thuật toán được đề xuất.

Từ khóa: điều khiển thích nghi bền vững (RAC), điều khiển trượt (SMC), mạng nơ ron hướng tâm (RBFNN)

1. ĐẶT VẤN ĐỀ

Điều khiển trượt (SMC) là một trong những bộ điều khiển có khả năng kháng nhiễu tốt và đảm bảo hệ thống kin ổn định tiệm cận [1]. Nhưng để thiết kế được bộ điều khiển trượt đòi hỏi chúng ta phải biết chính xác mô hình đối tượng. Để khắc phục nhược điểm này, bài báo đề xuất một phương pháp xấp xỉ hàm số bất định của đối tượng bằng một mạng nơ ron hướng tâm ba lớp (RBFNN) với các trọng số của mạng được chỉnh định on-line. Thuật toán đề xuất thích hợp cho lớp đối tượng bậc hai truyền ngược chặt bất định kiểu hàm số và có nhiều tác động. Thuật toán điều khiển đề xuất được kiểm chứng trên mô hình Robot 3 DOF. Cấu trúc bài báo gồm 4 phần: đặt vấn đề, Tổng hợp bộ điều khiển thích nghi bền vững, áp dụng cho mô hình robot ba bậc tự do và kết luận.

2. THUẬT TOÁN ĐIỀU KHIỂN THÍCH NGHI BỀN VỮNG

Bài toán thiết kế được phát biểu như sau: cho đối tượng có mô hình truyền ngược chặt

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1, x_2) + u(t) + n(t) \\ y = x_1 \end{cases} \quad (1)$$

trong đó $f(x_1, x_2)$ là hàm số bị chặn chưa biết của đối tượng và $n(r)$ là nhiễu bị chặn tác động từ bên ngoài vào hệ thống. x_1 và x_2 là các biến trạng thái quan sát được trực tiếp của đối tượng và hệ có điểm cân bằng tại gốc tọa độ. Tổng hợp luật điều khiển $u(r)$ đảm bảo tín hiệu ra bám tín hiệu đặt $w(r)$ với sai lệch bám tiến về 0.

2.1. Điều khiển trượt

Giả sử hàm $f(x_1, x_2)$ của đối tượng là xác định thì bài toán trên sẽ được giải quyết dễ dàng bằng phương pháp tổng hợp bộ điều khiển trượt. Theo [1], ta có mặt trượt S được định nghĩa như sau:

$$S(e, \dot{e}) = \lambda e + \dot{e} \quad \text{với } e = w - x_1, \dot{e} = \dot{w} - \dot{x}_2 \text{ và } \lambda > 0 \quad (2)$$

Tín hiệu điều khiển trượt được thiết kế dựa trên sự tồn tại một hàm Lyapunov $V(S)$ cho hệ kín.

Chọn hàm

$$V(S) = \frac{1}{2} S^2 \quad (3)$$

Từ đó ta có:

$$\dot{V}(S) = S \cdot \dot{S} = -SK \operatorname{sgn} S, \quad \text{với } K > 0 \quad (4)$$

Từ điều kiện (4) ta tổng hợp được tín hiệu điều khiển trượt u :

$$u = K \operatorname{sgn} S + \lambda \dot{e} + \ddot{w} - f(x_1, x_2) \quad (5)$$

nhưng do $f(x_1, x_2)$ là một hàm bất định nên ta không thể thực hiện được luật điều khiển (5). Bài báo đề xuất sử dụng RBFNN để xấp xỉ hàm bất định trên.

2.2. Xấp xỉ hàm bất định trên cơ sở FRBFNN

Để xấp xỉ hàm bất định $f(x_1, x_2)$ của (1) ta chuyển về dạng mô hình kết hợp giữa mô hình tuyến tính chuẩn điều khiển ổn định và phần bất định mới. Mô hình (1) được biểu diễn như sau:

$$\begin{cases} \dot{X} = \begin{bmatrix} 0 & 1 \\ -a_{21} & -a_{22} \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ F(X) \end{bmatrix} \\ y = x_1 \end{cases} \quad (6)$$

với a_{21} và a_{22} là các hệ số dương.

$$F(X) = a_{21}x_1 + a_{22}x_2 + f(x_1, x_2) = \begin{bmatrix} a_{21} & a_{22} \end{bmatrix} X + f(X) = D_m X + f(X) \quad (7)$$

Để xấp xỉ $F(X)$ ta sử dụng RBFNN biểu diễn ở Hình 1.

Mạng nơ ron được chọn bao gồm 3 lớp: lớp vào, lớp ra và lớp ẩn là các nơ ron hướng tâm RBF. Ta có hàm xấp xỉ $\hat{F}(x)$ được xác định bởi:

$$\hat{F}(X) = W^T \phi(X) \quad (8)$$

Với các hàm cơ sở được chọn [5]:

$$\phi_i(X) = \exp\left(-\frac{\|X - C_i\|^2}{b_i^2}\right) / \sum_{i=1}^m \exp\left(-\frac{\|X - C_i\|^2}{b_i^2}\right) \quad (9)$$

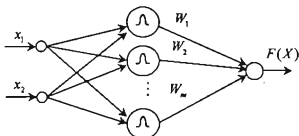
trong đó c_i và b_i là các hằng số dương tự chọn.

Đặt $A_m = \begin{bmatrix} 0 & 1 \\ -a_{21} & -a_{22} \end{bmatrix}$, $B_m = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, và $C_m = [1 \ 0]$ cấu trúc xấp xỉ hàm $F(x)$ được biểu diễn ở Hình 2.

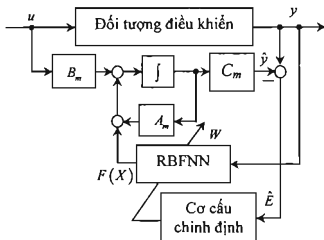
Như vậy, luật điều khiển trượt (5) được thực hiện bằng biểu thức sau:

$$u = K \operatorname{sgn} S + \lambda \dot{e} + \ddot{w} - \hat{f}(x_1, x_2) \quad (10)$$

$$\text{Với } \hat{f}(x_1, x_2) = \hat{F}(X) - [a_{21} \ a_{22}]X \quad (11)$$



Hình 1. RBFNN xấp xỉ hàm $F(X)$.



Hình 2. Cấu trúc bộ xấp xỉ hàm $F(X)$.

Để đảm bảo tính hội tụ của bộ xấp xỉ ta phải xác định được luật chỉnh định thích nghi véc tơ trọng số W đảm bảo sự tồn tại một hàm Lyapunov cho hệ kín, để véc tơ trọng số W tiến tới véc tơ trọng số tối ưu W^* . Hay nói cách khác là hàm $\hat{F}(X)$ tiến về hàm $F(X)$ với sai lệch ϵ nhỏ bao nhiêu tùy ý [2, 3, 4]. Từ Hình 2 ta có mô hình của bộ xấp xỉ:

$$\begin{cases} \dot{\hat{X}} = \begin{bmatrix} 0 & 1 \\ -a_{21} & -a_{22} \end{bmatrix} \hat{X} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ \hat{F}(\hat{X}) \end{bmatrix} \\ \hat{y} = C_m \hat{X} \end{cases} \quad (12)$$

Lấy phương trình (6) trừ đi phương trình (12) ta được sai số xấp xỉ

$$\dot{E} = \begin{bmatrix} 0 & 1 \\ -a_{21} & -a_{22} \end{bmatrix} E + \begin{bmatrix} 0 \\ F(X) - \hat{F}(\hat{X}) \end{bmatrix} = A_m E + \begin{bmatrix} 0 \\ e \end{bmatrix} = A_m E + \begin{bmatrix} 0 \\ \tilde{W}^T \phi(X) \end{bmatrix} \quad (13)$$

Ta định nghĩa một hàm Lyapunov:

$$V(E, \tilde{W}) = E^T P E + \tilde{W}^T \tilde{W} \quad (14)$$

Đạo hàm $V(.)$ theo thời gian ta được:

$$\dot{V}(E, \tilde{W}) = \dot{E}^T P E + E^T P \dot{E} + \dot{\tilde{W}}^T \tilde{W} + \tilde{W}^T \dot{\tilde{W}} \quad (15)$$

Thay (13) vào (15) ta được:

$$\dot{V}(E, \tilde{W}) = E^T (A_m^T P + P A_m) E + 2 \begin{bmatrix} 0 \\ \tilde{W}^T \phi(X) \end{bmatrix}^T P E + 2 \tilde{W}^T \dot{\tilde{W}} \quad (16)$$

Từ đó ta có:

$$\begin{aligned}\dot{V}(E, \tilde{W}) &= E^T (A_m^T P + P A_m) E + 2 \begin{bmatrix} 0 \\ \tilde{W}^T \phi(X) \end{bmatrix}^T \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + 2 \tilde{W}^T \tilde{W} \\ &= E^T (A_m^T P + P A_m) E + 2 \tilde{W}^T (\phi(X)(p_{12}e_1 + p_{22}e_2) + \dot{\tilde{W}})\end{aligned}\quad (17)$$

do A_m là ma trận Hurwitz nên dễ dàng chọn được ma trận P đối xứng xác định dương để có:

$$A_m^T P + P A_m = -Q \quad (18)$$

trong đó Q là ma trận đối xứng xác định dương. Với luật chỉnh định véc tơ trọng số:

$$\dot{\tilde{W}} = -\phi(X)(p_{12}e_1 + p_{22}e_2) \quad (19)$$

Thay (19) vào (17) ta có:

$$\dot{V}(E, \tilde{W}) = E^T (A_m^T P + P A_m) E = -E^T Q E < 0 \quad (20)$$

Vậy hàm $V(\cdot)$ định nghĩa ở (14) chính là hàm Lyapunov của hệ. Cấu trúc hệ thống điều khiển được biểu diễn ở Hình 3.

3. ÁP DỤNG CHO ROBOT BA BẬC TỰ DO

3.1. Thiết kế bộ điều khiển

Theo [6] từ mô hình động lực học của robot n bậc tự do:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F_d(\dot{q}) + \tau_d = \tau \quad (21)$$

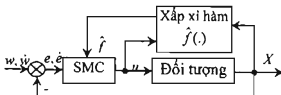
với $q \in R^n$, $\dot{q} \in R^n$, $\ddot{q} \in R^n$ là véc tơ góc, tốc độ và gia tốc của n khớp. Ma trận quán tính $H(q)$ là ma trận đối xứng xác định dương và biết trước do vậy có thể biểu diễn robot n bậc tự do bằng n hệ con truyền ngược chặt sau:

$$\begin{cases} q_i = x_{i1} \\ \dot{q}_i = x_{i2} \\ \ddot{q}_i = \dot{x}_{i2} = f_i(x_i) + u_i \end{cases} \quad (22)$$

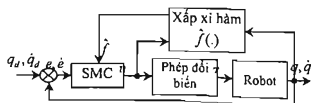
với $x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \end{bmatrix}$; $i = 1, 2, \dots, n$; $f_i(x_i)$ là các hàm trơn chưa biết và bị chặn. Quỹ đạo đặt $(q_d, \dot{q}_d)$ cho trước và phép đổi biến:

$$\begin{cases} H^{-1}(q)\tau = g(q)\tau = u; \\ \tau = H(q)u = g^{-1}(q)u; \quad u \in R^n \end{cases} \quad (23)$$

Mô hình robot biểu diễn ở dạng (22) cho phép sử dụng phương pháp thiết kế bộ điều khiển trượt mờ thích nghi bền vững (SMCFNN) có cấu trúc như hình sau:



Hình 3. Cấu trúc bộ điều khiển SMCNN.



Hình 4. Cấu trúc bộ điều khiển SMCFNN.

Thiết kế bộ điều khiển trượt

Mô hình (22) có $f_1(\cdot)$ biết trước. Thiết kế bộ điều khiển trượt cho khớp 1.

Chọn mặt trượt:

$$S_1(e_1) = \lambda e_1 + \dot{e}_1; \lambda > 0 \quad (24)$$

trong đó: $e_1 = q_{1d} - q_1$ với q_{1d} là quỹ đạo và quỹ đạo đặt của khâu 1

Lấy đạo hàm (24) ta có:

$$\frac{dS_1(e_1)}{dt} = \lambda \dot{e}_1 + \ddot{e}_1 = \lambda(\dot{q}_{1d} - \dot{q}_1) + \ddot{q}_{1d} - \ddot{q}_1 = \lambda\dot{q}_{1d} + \ddot{q}_{1d} - \lambda\dot{q}_1 - (f_1 + u_1) \quad (25)$$

kết hợp điều kiện trượt ta có:

$$[\lambda\dot{q}_{1d} + \ddot{q}_{1d} - \lambda\dot{q}_1 - (f_1 + u_1)]K \operatorname{sgn}(S_1) < 0 \Rightarrow \lambda\dot{q}_{1d} + \ddot{q}_{1d} - \lambda\dot{q}_1 - (f_1 + u_1) = -K \operatorname{sgn}(S_1) \quad (26)$$

ta có bộ điều khiển trượt cho khớp thứ nhất:

$$u_1 = K \operatorname{sgn}(S_1) + \lambda\dot{q}_{1d} + \ddot{q}_{1d} - \lambda\dot{q}_1 - f_1 \quad (27)$$

Tương tự cho khớp thứ hai ta có:

$$u_2 = K \operatorname{sgn}(S_2) + \lambda\dot{q}_{2d} + \ddot{q}_{2d} - \lambda\dot{q}_2 - f_2 \quad (28)$$

với $S_2(e_2) = \lambda e_2 + \dot{e}_2$; $e_2 = q_{2d} - q_2$ và q_{2d} là quỹ đạo và quỹ đạo đặt của khâu 2.

Bộ điều khiển trượt cho khớp thứ 3:

$$u_3 = K \operatorname{sgn}(S_3) + \lambda\dot{q}_{3d} + \ddot{q}_{3d} - \lambda\dot{q}_3 - f_3 \quad (29)$$

với $S_3(e_3) = \lambda e_3 + \dot{e}_3$; $e_3 = q_{3d} - q_3$ và q_{3d} là quỹ đạo và quỹ đạo đặt của khâu 3.

Xấp xỉ hàm phi tuyến bất định

Khi có được $F_j(q, \dot{q})$ ta dễ dàng xác định $\hat{f}_j(q, \dot{q})$ theo biểu thức sau:

$$\hat{f}_j(q, \dot{q}) = F_j(q, \dot{q}) - a_{21}q_j - a_{22}\dot{q}_j \quad (30)$$

trong đó $a_{21} > 0; a_{22} > 0$ là các tham số được chọn trước; $j = 1, 2, 3$.

Để xấp xỉ hàm phi tuyến bất định ta sử dụng mạng mờ ron RBF ta có:

$$F_j(q, \dot{q}) = \sum_{i=1}^m W_i^* \phi_i(q, \dot{q}) + \varepsilon \quad (31)$$

trong đó $|\varepsilon| < \varepsilon_M$, ε_M là sai số cho trước; $\phi_i(q, \dot{q})$ là các hàm cơ sở xuyên tâm và được chọn:

$$\phi_i(q, \dot{q}) = \exp \left(-\frac{\| [q, \dot{q}]^T - C_i \|^2}{b_i^2} \right) / \sum_{i=1}^m \exp \left(-\frac{\| [q, \dot{q}]^T - C_i \|^2}{b_i^2} \right) \quad (32)$$

với C_i là véc tơ 2 chiều biểu diễn tâm của hàm cơ sở thứ i , b_i biểu diễn độ trải rộng của hàm cơ sở. Ở đây các trọng số lí tưởng W_i^* không biết trước. Nhiệm vụ đặt ra là xác định các trọng số đánh giá $\hat{W}_i$, theo đó đánh giá hàm phi tuyến

$$\hat{F}(q, \dot{q}) : \hat{F}(q, \dot{q}) = \sum_{i=1}^m \hat{W}_i \phi_i(q, \dot{q}) \quad (33)$$

Quá trình xác định $\hat{W}_i$ là quá trình học của mạng, chính là quá trình hiệu chỉnh các trọng số $\hat{W}_i$ lớp ra của mạng RBF.

Sai lệch của trọng số đánh giá so với trọng số lí tưởng sẽ là:

$$\tilde{W}_i = \hat{W}_i - W_i^* \quad (34)$$

Từ (31), (33) và (34) ta có:

$$F(q, \dot{q}) = \hat{F}(q, \dot{q}) + \varepsilon^* ; \varepsilon^* = \varepsilon - \sum_{i=1}^m \tilde{W}_i \phi_i(q, \dot{q}) \quad (35)$$

Khi sai lệch của các trọng số đánh giá so với các trọng số lí tưởng tương ứng bị triệt tiêu $\tilde{W}_i \rightarrow 0; i = 1, 2, \dots, m$ thì $\varepsilon = \varepsilon^*$. Luật cập nhật trọng số của mạng:

$$\dot{\tilde{W}}_i = -\phi_i(q, \dot{q})(p_{11}e_1 + p_{12}e_2) \quad (36)$$

với $e_1 = q_{1,d} - q_1; e_2 = \dot{q}_{1,d} - \dot{q}_1$. Luật hiệu chỉnh này được xác định sao cho $W_i \rightarrow W_i^*, \tilde{W}_i \rightarrow 0, \hat{F}(q, \dot{q}) \rightarrow F(q, \dot{q})$. Quá trình hiệu chỉnh được thực hiện trên cơ sở sử dụng véc tơ sai lệch $E(t)$ và đảm bảo để $E(t) \rightarrow 0$.

3.2. Mô phỏng kiểm chứng bộ điều khiển

- Robot Scara 3 bậc tự do bao gồm 2 khớp quay và 1 khớp tịnh tiến được thể hiện như hình 5.

Thông số của robot được cho như sau:

$$m_1 = 3,27(kg); m_2 = 2,93(kg); m_3 = 2,13(kg); l_1 = 0,45(m); l_2 = 0,45(m); l_3 = 0,565(m)$$

- Nhiều và lực ma sát tác động lên các khớp dạng đặc tính được biểu thị như hình 8.

+ Nhiều tải trong khoảng

$$5(s) : t = 1 + 2(s); t = 3 + 4(s) \text{ là: } \tau_d = [10 \quad 10 \quad 10]^T \quad (37)$$

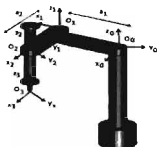
+ Lực ma sát nhớt và khô:

$$F_{ms_i} = 3 * \dot{q}_i + 2 * \text{sgn}(\dot{q}_i) \quad (38)$$

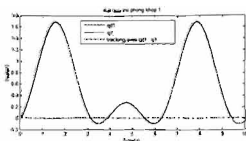
- Quỹ đạo mẫu cho 3 khớp như sau:

$$q_{d1} = (\sin(t) + e^{\sin(0.6t)}) \sin(0.6t)(rad); q_{d2} = (\sin(t) + e^{\sin(0.5t)}) \sin(0.5t)(rad); q_{d3} = 0.1 \sin(t)(m) \quad (39)$$

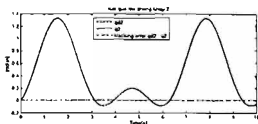
- Kết quả mô phỏng



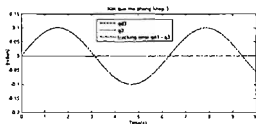
Hình 5. Mô hình Robot Scara 3 bậc tự do.



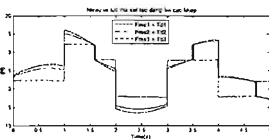
Hình 6. Kết quả mô phỏng bám quỹ đạo đặt và sai lệch bám của khớp 1.



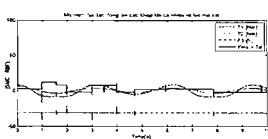
Hình 7. Kết quả mô phỏng bám quỹ đạo đặt và sai lệch bám của khớp 2.



Hình 8. Kết quả mô phỏng bám quỹ đạo đặt và sai lệch bám của khớp 3.



Hình 9. Dạng đặc tính của lực ma sát và nhiễu tác động lên các khớp.



Hình 10. Kết quả mô phỏng mô men và lực tác động lên các khớp khi chịu ảnh hưởng của lực ma sát và nhiễu.

4. KẾT LUẬN

Qua kết quả mô phỏng cho thấy với thuật toán điều khiển trượt mờ thích nghi được đề xuất cho chất lượng bám quỹ đạo đặt tốt, sai lệch bám nhỏ. Mô men và lực tác động lên các khớp đảm bảo bù được nhiễu và lực ma sát tác động lên hệ thống do đó nó có thể áp dụng cho các đối tượng robot trong thực tế.

TÀI LIỆU THAM KHẢO

1. Wilfrid Preruquetti, Jean Pierre Barbot (Eds.) - Sliding Mode Control in Engineering, Copyright by Marcel Dekker, Inc. All Rights Reserved, Printed USA, 2002.
2. Hornik K., Stinchcombe M., and White H. - Multilayer feedforward networks are universal approximators, Neural Network (2) (1989) 359-365.
3. Gorn J. B., Yu D. L. - Selecting radial basis function networks centers with recursive orthogonal least squares training, IEEE Transaction on Neural Networks 11 (2) (2000) 306-314.

4. Panchapakesan C., Ralph D., and Palanisami M. - Effects of moving the center in an RBF Network. Proceedings of the 1998 IEEE World Congress on Computational Intelligence Neural Networks 2 (1998)1256-1269.
5. Huang S. N., Tan K. K., Lee T. H. - A combined PID/Addaptive controller for a class of nonlinear Systems, Automatica 37 (2001) 611-618.
6. Nguyen Thai-Huu, Minh Phan-Xuan, Son Hoang-Minh, Dan Nguyen-Cong, Quyet Ho-Gia - Robust adaptive control of robots using neural network and sliding mode control, 2013 International Conference on Control, Automation and Information Sciences, ICCAIS-2013, IEEE Nha Trang 11/2013.

ABSTRACT

ROBUST ADAPTIVE NEURAL NETWORKS SLIDING MODE CONTROL FOR THREE-FREEDOM ROBOT

Thai Huu Nguyen^{1,*}, Phan Xuan Minh², Nguyen Cong Khoa²

¹ *Vinh University of Technology Education,*

² *Department of Automatic Control, School of Electrical Engineering,
Hanoi University of Science and Technology*

*Email: thainguyenktv@yahoo.com

This paper presents an algorithm to design a robust adaptive controller for strictly parameter feedback plant with function uncertainty. The control algorithm is synthesized on sliding mode control principle. The function uncertainty of the plant is approximated by a two layer radial basis function neural networks, the weigh coefficients are trained on-line. The proposed controller is used to control a three-freedom robot model. The simulation results show the advances and the application ability in practice of the proposed control algorithm.

Keywords: robust adaptive control (RAC), sliding mode control (SMC), radial basis function neural network (RBFNN).