



DOI:10.22144/ctu.jvn.2018.120

ĐÁNH GIÁ HIỆU QUẢ CỦA GIẢI THUẬT QUASI-NEWTON TRONG ĐIỀU KHIỂN TRƯỢT THÍCH NGHI SỬ DỤNG MẠNG NƠ-RON RBF

Phạm Thanh Tùng^{1*}, Nguyễn Đình Tứ², Lê Thị Kiều Mai¹, Nguyễn Hứa Duy Khang²,
 Đồng Văn Hương³ và Nguyễn Chí Ngôn²

¹Trường Đại học Sư phạm Kỹ thuật Vĩnh Long

²Khoa Công nghệ, Trường Đại học Cần Thơ

³Trường Đại học Giao thông vận tải thành phố Hồ Chí Minh

*Người chịu trách nhiệm về bài viết: Phạm Thanh Tùng (email: tungpt@vnlute.edu.vn)

Thông tin chung:

Ngày nhận bài: 18/05/2018

Ngày nhận bài sửa: 31/07/2018

Ngày duyệt đăng: 29/10/2018

Title:

Performance evaluation of the Quasi-Newton algorithm in adaptive sliding mode control using radial basis function neural networks

Từ khóa:

Điều khiển trượt thích nghi, giải thuật Quasi-Newton, huấn luyện trực tuyến, mạng nơ-ron RBF, robot di động đa hướng

Keywords:

Adaptive sliding mode control, Omni-directional mobile robot, online training, Quasi-Newton algorithm, radial basis function neural networks

ABSTRACT

This paper is aimed to evaluate the performance of the Quasi-Newton in online training of an adaptive sliding mode control using radial basis function neural networks. The controller is applied in trajectory tracking of an Omni-directional mobile robot. The sliding mode control plays a role of tracking the trajectories of the robot reaching the references. The radial basis function neural networks are used to estimate nonlinear functions in the sliding mode control law which is calculated based on the Lyapunov stability theory adapting to control conditions. The simulation results in MATLAB/SIMULINK show that the Quasi-Newton algorithm applying in the proposed controller is positive effect; the responses of the robot converge to references without steady-state error; the overshoot is only 0.442 (%), and the mean square error approximates 3.48×10^{-4} .

TÓM TẮT

Bài báo này nhằm đánh giá hiệu quả của giải thuật Quasi-Newton trong huấn luyện trực tuyến bộ điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF (radial basis function). Giải thuật này được ứng dụng trong điều khiển bám quỹ đạo robot di động đa hướng. Bộ điều khiển trượt đóng vai trò điều khiển robot bám quỹ đạo tham khảo. Mạng nơ-ron RBF được sử dụng để ước lượng các hàm phi tuyến trong luật điều khiển trượt, được tính toán dựa trên lý thuyết ổn định Lyapunov thích nghi với các điều kiện thực tế. Kết quả mô phỏng với MATLAB/SIMULINK cho thấy giải thuật Quasi-Newton áp dụng trong bộ điều khiển đề xuất đạt hiệu quả tốt; sai số xác lập của các đáp ứng bị triệt tiêu; độ vọt lố đạt 0.442 (%) và sai số trung bình bình phương xấp xỉ 3.48×10^{-4} .

Trích dẫn: Phạm Thanh Tùng, Nguyễn Đình Tứ, Lê Thị Kiều Mai, Nguyễn Hứa Duy Khang, Đồng Văn Hương và Nguyễn Chí Ngôn, 2018. Đánh giá hiệu quả của giải thuật Quasi-Newton trong điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF. Tạp chí Khoa học Trường Đại học Cần Thơ. 54(7A): 27-34.

1 GIỚI THIỆU

Mạng nơ-ron RBF (radial basis function) được giới thiệu năm 1988 và gần đây nhận được nhiều sự chú ý do khả năng tổng quát hóa tốt và cấu trúc đơn giản, tính toán nhanh và hiệu quả (Liu, 2013). Nó có thể cải thiện hiệu quả điều khiển ngay cả khi có sự bất định lớn của hệ thống. Các nhà nghiên cứu đã đề xuất nhiều phương pháp để huấn luyện mạng RBF như: Gradient Descent, Newton’s method, Conjugate Gradient, Quasi-Newton, Levenberg Marquardt (Burney *et al.*, 2007). Trong đó, giải thuật Quasi-Newton đã được nhiều tác giả nghiên cứu. Likas and Stafylopatis (2000) đã sử dụng kỹ thuật tối ưu Quasi-Newton để cực tiểu hóa sai số trong huấn luyện mạng nơ-ron ngẫu nhiên. Kết quả mô phỏng cho thấy rằng giải thuật Quasi-Newton vượt trội hơn so với giải thuật Gradient Descent (Relhan and Jain, 2012). Giải thuật Quasi-Newton tỏ ra hiệu quả trong huấn luyện mạng nơ-ron 3 lớp ẩn sử dụng trong kỹ thuật nén ảnh (Omer Mahmoud *et al.*, 2007), phân loại Hematoma ảnh CT não (Sharma and Venugopalan, 2014). Ngoài ra, nghiên cứu của Mukherjee and Routroy (2012) cũng cho thấy giải thuật Quasi-Newton đáp ứng tốt yêu cầu huấn luyện bộ điều khiển mạng nơ-ron cho quá trình mài. Tuy nhiên, các nghiên cứu về việc ứng dụng giải thuật Quasi-Newton trong huấn luyện trực tuyến mạng nơ-ron RBF cho bộ điều khiển trượt thích nghi vẫn còn nhiều hạn chế.

Bài báo này đề xuất sử dụng giải thuật Quasi-Newton để huấn luyện trực tuyến mạng nơ-ron RBF trong bộ điều khiển trượt thích nghi áp dụng để điều khiển robot di động đa hướng. Đây là một loại robot holonomic có thể di chuyển dễ dàng trong những không gian nhỏ, hẹp do khả năng di chuyển một cách linh hoạt, vừa quay vừa tịnh tiến đồng thời và độc lập (Djebrani *et al.*, 2012). Khả năng vốn có của robot đã làm cho nó được nghiên cứu rộng rãi và được ứng dụng trong nhiều môi trường năng động.

2 GIẢI THUẬT QUASI-NEWTON VỚI BFGS

BFGS (Broyden-Fletcher-Goldfarb-Shanno) là một trong các phương pháp Quasi-Newton sử dụng các phép xấp xỉ thay cho các phương pháp ma trận nghịch đảo Hessian. Nếu ta có bài toán cực tiểu (Sadeghi *et al.*, 2016) theo (1):

$$\text{Min } f(x), x \in R^n \tag{1}$$

Cho $f(x^*)$ là cực tiểu, quá trình lặp lại để tìm kiếm x^* như (2):

$$X_{k+1} = X_k - \alpha_k \Xi_k \nabla f^T(x_k) \tag{2}$$

Trong đó, Ξ_k là ma trận đối xứng $n \times n$ và α_k là tốc độ học sẽ được tìm để cực tiểu $f(x_{k+1})$. Ξ_k là ma trận Hessian mà phương pháp Newton cần tìm (khi $\Xi_k = I$ thì ta có phương pháp Gradient Descent).

Giả sử $f \in R^n$ chứa các đạo hàm từng phần bậc hai liên tục, ta định nghĩa như (3):

$$g_k = \nabla f^T(x_k), p_k = X_{k+1} - X_k \tag{3}$$

Gọi H_k là ma trận xấp xỉ ma trận Hessian trong phép lặp thứ k . Bắt đầu từ mỗi ma trận đối xứng hằng số dương H_0 , mỗi điểm x_0 và $k = 0$ được thực hiện theo 3 bước:

Bước 1: xét $d_k = -H_k^{-1} g_k$

Bước 2: cực tiểu $f(x_k + \alpha d_k)$ với $\alpha \geq 0$

Ta có $X_{k+1}, P_k = \alpha_k d_k, g_{k+1}$

Bước 3 : xét $q_k = g_{k+1} - g_k$

$$\text{Và } H_{k+1} = H_k + \frac{p_k p_k^T}{p_k^T q_k} - \frac{H_k q_k q_k^T H_k}{q_k^T H_k q_k} \tag{4}$$

Sau đó cập nhật k và quay về Bước 1.

Nếu H_k là giá trị thực thì H_{k+1} cũng là một giá trị thực và trình tự của thuật toán này hội tụ (Sadeghi *et al.*, 2016). Một phương pháp mới để cập nhật cho H được BFGS đề xuất như (5):

$$H_{k+1} = H_k + \left(1 + \frac{q_k^T H_k q_k}{q_k^T p_k} \right) \frac{p_k p_k^T}{p_k^T q_k} - \frac{p_k q_k^T H_k + H_k q_k p_k^T}{q_k^T p_k} \tag{5}$$

3 ĐIỀU KHIỂN TRƯỢT THÍCH NGHI SỬ DỤNG MẠNG NƠ-RON RBF ROBOT DI ĐỘNG ĐA HƯỚNG

3.1 Mô hình toán học của robot

Phương trình động lực học robot được trình bày như (6) (Watanabe, 1998) :

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\phi}_w \end{bmatrix} = \begin{bmatrix} a_1 & -a_2 \phi_d & 0 \\ a_2 \phi_d & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi}_w \end{bmatrix} + \begin{bmatrix} b_1 \gamma_1 & b_1 \gamma_2 & 2b_1 \cos \phi \\ b_1 \gamma_3 & b_1 \gamma_4 & 2b_1 \sin \phi \\ b_2 & b_2 & b_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} D_{f_x} \\ D_{f_y} \\ D_{f_\phi} \end{bmatrix} \tag{6}$$

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

Với $D_f = [D_{fx} \ D_{fy} \ D_{f\phi}]^T$ là nhiều hệ thống chưa biết.

$$a_1 = \frac{-3c}{(3I_w + 2Mr^2)}; a_2 = \frac{2Mr^2}{(3I_w + 2Mr^2)};$$

$$a_3 = \frac{-3cL^2}{(3I_w L^2 + I_v r^2)}$$

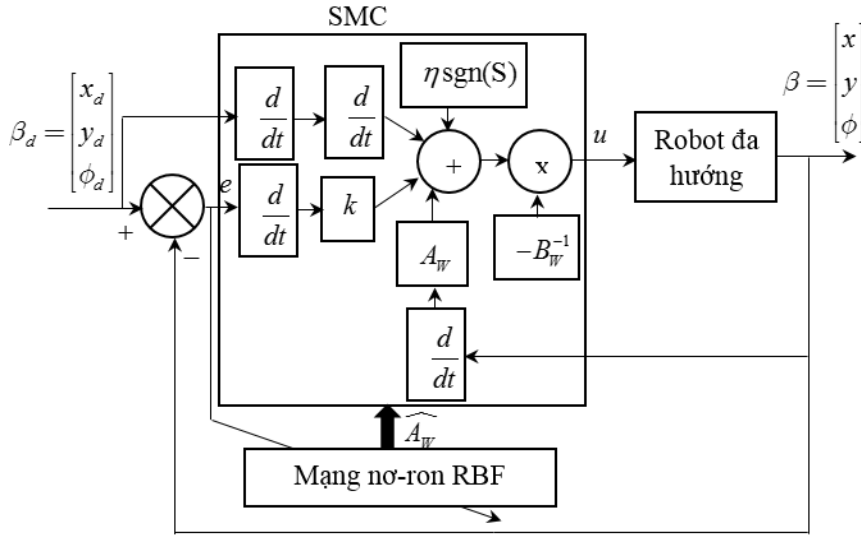
$$b_1 = \frac{kr}{(3I_w + 2Mr^2)}; b_2 = \frac{krL}{(3I_w + I_v r^2)}$$

$$\gamma_1 = -\sqrt{3} \sin \phi - \cos \phi; \gamma_2 = \sqrt{3} \sin \phi - \cos \phi; \gamma_3 = \sqrt{3} \cos \phi - \sin \phi; \gamma_4 = -\sqrt{3} \cos \phi - \sin \phi$$

Trong đó, L : khoảng cách giữa bất kỳ bộ phận và tâm trọng lực của robot (m); c : hệ số ma sát nhớt cho bánh xe (kgm^2/s); r : bán kính của mỗi bánh xe (m); I_w : mô-men quán tính của mỗi bánh xe quanh trục lái (kgm^2); I_v : mô-men quán tính của robot (kgm^2); k : hệ số truyền động; M : mô-men quán tính của hòn bi (kgm^2).

3.2 Điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF

Cấu trúc của bộ điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF (ASMC-RBF: adaptive sliding mode control – RBF) được mô tả như Hình 1.



Hình 1: Cấu trúc bộ điều khiển ASMC-RBF

Sai số được định nghĩa như (7):

$$e = \beta - \beta_d = \begin{bmatrix} e_x \\ e_y \\ e_\phi \end{bmatrix} \quad (7)$$

Trong đó $\beta = [x_w \ y_w \ \phi_w]^T$ và $\beta_d = [x_d \ y_d \ \phi_d]^T$

Mặt trượt được mô tả như (8) :

$$S = \begin{bmatrix} (\dot{x}_w - \dot{x}_d) + k_x (x_w - x_d) \\ (\dot{y}_w - \dot{y}_d) + k_y (y_w - y_d) \\ (\dot{\phi}_w - \dot{\phi}_d) + k_\phi (\phi_w - \phi_d) \end{bmatrix} \quad (8)$$

$$= \begin{bmatrix} \dot{e}_x + k_x e_x \\ \dot{e}_y + k_y e_y \\ \dot{e}_\phi + k_\phi e_\phi \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \\ S_\phi \end{bmatrix}$$

Trong đó, $k = \text{diag}([k_x \ k_y \ k_\phi])$ là hằng số dương. Mặt trượt và hàm thông số nhiều được biểu diễn trong không gian vector lần lượt $S = [S_x \ S_y \ S_\phi]^T, d(t)$.

Bộ điều khiển trượt thích nghi được trình bày nhằm giải quyết các nguyên nhân gây ra bởi các thông số bất định của hệ thống.

Luật điều khiển trượt được định nghĩa như (9) (Yang and Cheng, 2013):

$$U = -B_w^{-1} \{A_w \dot{\beta} + (\ddot{\beta}_d + k\dot{e}) + \eta \text{sgn}(S)\} \quad (9)$$

Trong đó, $\eta = \text{diag}([\eta_x, \eta_y, \eta_\phi])$ là ma trận xác định dương.

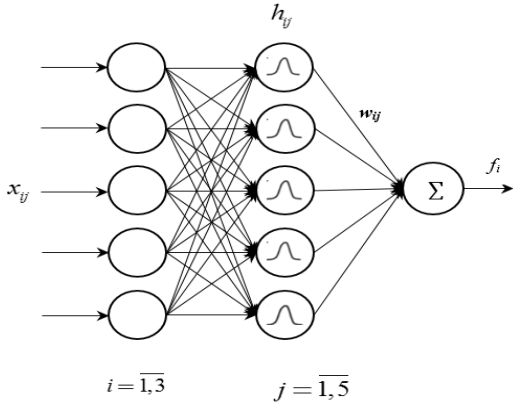
Trong phần này, nghiên cứu sử dụng giải thuật Quasi-Newton để xấp xỉ A_w . Đây là ma trận chứa

các thông số đặc trưng của robot như: bán kính của mỗi bánh xe (r), mô-men quán tính của robot (I_v) và mô-men quán tính của hòn bi (M).

Thuật toán của mạng nơ-ron RBF như (10) (Liu, 2013):

$$h_{ij} = \exp\left(-\frac{\|X_i - c_{ij}\|^2}{2b_{ij}^2}\right) \Big|_{i=\overline{1,3}; j=\overline{1,5}} \quad (10)$$

Cấu trúc mạng RBF được sử dụng để xấp xỉ thành phần f_i trong ma trận A_W như Hình 2:



Hình 2: Cấu trúc mạng nơ-ron RBF xấp xỉ thành phần f_i

Trong đó,

$$X_i = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} e(1) & \dot{e}(1) & \beta_d(1) & \dot{\beta}_d(1) & \ddot{\beta}_d(1) \\ e(2) & \dot{e}(2) & \beta_d(2) & \dot{\beta}_d(2) & \ddot{\beta}_d(2) \\ e(3) & \dot{e}(3) & \beta_d(3) & \dot{\beta}_d(3) & \ddot{\beta}_d(3) \end{bmatrix} \quad (11)$$

$$h_{ij} \Big|_{i=1,2,3} = [h_{i1} \quad h_{i2} \quad h_{i3} \quad h_{i4} \quad h_{i5}] \quad (12)$$

$$w_{ij} \Big|_{i=1,2,3} = [w_{i1} \quad w_{i2} \quad w_{i3} \quad w_{i4} \quad w_{i5}] \quad (13)$$

$$\text{Ngõ ra } f_i \text{ như (14): } f_i = w_{ij}^T h_{ij} \quad (14)$$

Kết quả xấp xỉ ma trận A_W sử dụng mạng nơ-ron RBF dựa trên giải thuật Quasi-Newton như (15):

$$\widehat{A}_W = \begin{bmatrix} w_{1j}^T h_{1j} & -w_{2j}^T \phi_d h_{2j} & 0 \\ w_{2j}^T h_{2j} & w_{1j}^T h_{1j} & 0 \\ 0 & 0 & w_{3j}^T h_{3j} \end{bmatrix} \quad (15)$$

Luật điều khiển trượt thích nghi dùng mạng nơ-ron RBF (ASMC-RBF) như (16):

$$U = -B_w^{-1} \left\{ \widehat{A}_w \dot{\beta} + (\dot{\beta}_d + k\dot{e}) + \eta \text{sign}(S) \right\} \quad (16)$$

Khi quỹ đạo thực tế của robot lệch khỏi quỹ đạo tham khảo do tác động của các điều kiện thực tế như ma sát mặt đường, mô-men quán tính thay đổi,... thì sai số (7) thay đổi. Khi đó, mạng RBF sẽ được tự động cập nhật bằng giải thuật Quasi-Newton, dẫn đến \widehat{A}_w cũng thay đổi theo sao cho sai số (7) đạt giá trị tối thiểu. Ta nói luật điều khiển (16) thích nghi theo điều kiện hoạt động của robot.

Để chứng minh tính ổn định của luật điều khiển, ta cần chứng minh mặt trượt hội tụ về 0 theo Lyapunov. Hàm Lyapunov được định nghĩa như (17):

$$V = \frac{1}{2} S^T S > 0 \quad (17)$$

Lấy đạo hàm của V ta có (18):

$$\begin{aligned} \dot{V} &= S^T \left(-\widehat{A}_w \dot{\beta} - \eta \text{sign}(S) \right) \\ &= S^T \left(-\widetilde{\Gamma} \dot{\beta} - \eta \text{sign}(S) \right) \\ &= -S^T \left(\widetilde{\Gamma} \dot{\beta} + \eta \text{sign}(S) \right) \end{aligned} \quad (18)$$

Yang and Cheng (2013) đã chứng minh $\dot{V} \leq 0$. Trong đó, ma trận η là ma trận đối xứng xác định dương. Thêm vào đó, sai số $e(t)$ sẽ hội tụ về 0 dẫn theo $S(t) \rightarrow 0$ khi $t \rightarrow \infty$. Vì thế, $e(t), \dot{e}(t) \rightarrow 0$ khi $t \rightarrow \infty$.

3.3 Thông số và kết quả mô phỏng

Thông số mô phỏng

Thông số của robot được trình bày như Bảng 1:

Bảng 1: Các thông số của robot di động đa hướng (Watanabe, 1998)

Ký hiệu	Ý nghĩa	Giá trị	Đơn vị
I_v	Mô-men quán tính của robot	11.25	kgm^2
M	Moment quán tính của hòn bi	9.4	kgm^2
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_ω	Mô-men quán tính của bánh xe	0.02108	kgm^2
r	Bán kính mỗi bánh xe	0.0245	m

Thông số của bộ điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF được khởi tạo nằm trong

giới hạn giá trị của X_i , được trình bày trong Bảng 2 và 3.

Bảng 2: Các tham số của bộ điều khiển trượt thích nghi

Ý nghĩa	Ký hiệu	Giá trị
Thông số trượt	$\begin{bmatrix} \eta_x & 0 & 0 \\ 0 & \eta_y & 0 \\ 0 & 0 & \eta_\phi \end{bmatrix}$	$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$
	$\begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\phi \end{bmatrix}$	$\begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 58 \end{bmatrix}$

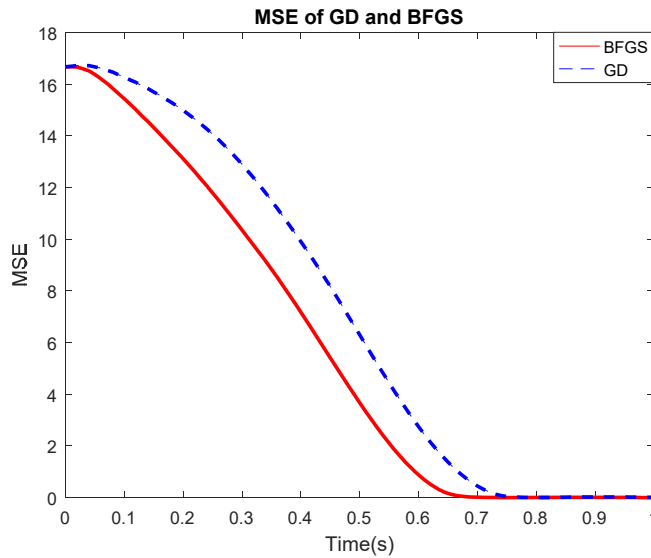
Bảng 3: Các tham số trong mạng nơ-ron RBF

Ký hiệu	Ý nghĩa	Giá trị
c_{ij}	Véc-tơ trung tâm của mạng RBF	$\begin{bmatrix} -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \end{bmatrix}$
b_j	Ngưỡng kích hoạt	$[1 \ 1 \ 1 \ 1 \ 1]^T$
$[\mathbf{w}_{1j} \ \mathbf{w}_{2j} \ \mathbf{w}_{3j}]^T$	Véc-tơ trọng số khởi tạo ban đầu	$\begin{bmatrix} -0.1347 & 0.7094 & 0.4588 \\ 0.2546 & 0.2970 & 0.9450 \\ 0.9174 & 0.7919 & -0.0617 \\ 0.8973 & 0.2517 & -0.5407 \\ 0.6460 & -0.0970 & 0.4458 \end{bmatrix}^T$
i	Số mạng RBF	3
j	Số nơ-ron lớp ẩn của mỗi mạng	5

Kết quả mô phỏng

Kết quả huấn luyện trực tuyến mạng nơ-ron RBF với giải thuật Quasi-Newton được trình bày như Hình 3 với MSE (Mean Square Error) được so sánh với giải thuật Gradient Descent. Qua Hình 3 ta thấy rằng, giá trị MSE của giải thuật Quasi-

Newton (5.89×10^{-4}) nhỏ hơn MSE của giải thuật Gradient Descent (4.1×10^{-3}). Ngoài ra, Hình 3 còn cho thấy thời gian huấn luyện mạng RBF bằng giải thuật BFGS nhanh hơn giải thuật Gradient Descent truyền thống. Điều này có ý nghĩa lớn trong việc cải thiện đáp ứng của bộ điều khiển, đặc biệt trong kỹ thuật huấn luyện trực tuyến bộ điều khiển.



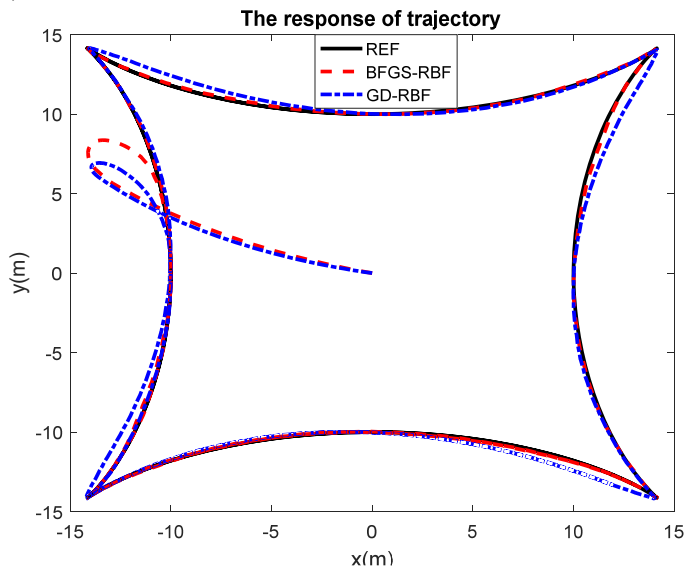
Hình 3: So sánh MSE của Quasi-Newton và Gradient Descent

Giải thuật huấn luyện trực tuyến mạng nơ-ron RBF với Quasi-Newton được ứng dụng điều khiển bám quỹ đạo robot di động đa hướng với quỹ đạo mong muốn được trình bày như (19) và (20):

$$\begin{aligned} x_d &= -\sin(2t) \left(20(\cos(2t))^2 + 10 \right) (m) \\ y_d &= \cos(2t) \left(20(\sin(2t))^2 + 10 \right) (m) \\ \phi_d &= -0.2t \text{ (rad / s)} \end{aligned} \quad (19)$$

$$\begin{aligned} x_d &= 5 \cos(2\pi t) (m) \\ y_d &= 5 \sin(2\pi t) (m) \\ \phi_d &= -0.2\pi t \text{ (rad / s)} \end{aligned} \quad (20)$$

Đáp ứng bộ điều khiển BFGS-ASMC-RBF với quỹ đạo tham chiếu (19) được trình bày trong Hình 4.

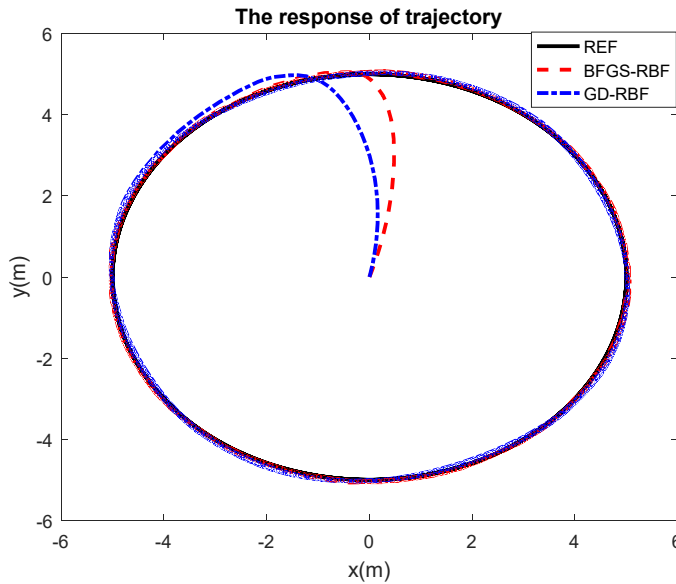


Hình 4: Đáp ứng quỹ đạo bộ điều khiển theo (19)

Đáp ứng bộ điều khiển BFGS-ASMC-RBF với quỹ đạo tham chiếu (20) được trình bày trong Hình 5.

Kết quả mô phỏng cho thấy đáp ứng bộ điều

khiển BFGS-ASMC-RBF tốt hơn bộ điều khiển trượt thích nghi với mạng nơ-ron RBF được huấn luyện trực tuyến sử dụng giải thuật Gradient Descent (GD-ASMC-RBF) được trình bày chi tiết trong Bảng 4.



Hình 5: Đáp ứng quỹ đạo bộ điều khiển theo (20)

Bảng 4: Các chỉ tiêu chất lượng giữa Quasi-Newton và Gradient Descent

Đáp ứng	Gradient Descent			Quasi-Newton		
	Thời gian tăng (ms)	Độ vọt lố (%)	Sai số xác lập	Thời gian tăng (ms)	Độ vọt lố (%)	Sai số xác lập
x	291.161	0.505	1.137	292.796	0.442	0.0552
y	292.845	1.368	0.0722	292.654	0.469	0.0434

Hiệu suất của giải thuật Quasi-Newton với tiêu chí RMS (Root Mean Square) của từng đáp ứng x,

y và ϕ so với giải thuật Gradient Descent được trình bày chi tiết trong Bảng 5.

Bảng 5: So sánh RMS và MSE của Quasi-Newton và Gradient Descent

Đáp ứng	Gradient Descent		Quasi-Newton	
	RMS	MSE	RMS	MSE
x	3.539	3.96×10^{-4}	3.471	5.46×10^{-5}
y	3.537	2.6×10^{-3}	3.538	9.4×10^{-4}
ϕ	1.815	5.53×10^{-5}	1.815	4.98×10^{-5}
Trung bình	2.9637	1×10^{-3}	2.9413	3.48×10^{-4}

Qua Bảng 5, tiêu chí RMS của từng đáp ứng x, y và ϕ của giải thuật Quasi-Newton tương ứng là 3.471, 3.538 và 1.815; tiêu chí MSE là 5.46×10^{-5} , 9.4×10^{-4} và 4.98×10^{-5} vượt trội hơn so với giải thuật Gradient Descent. Điều này có ý nghĩa quan trọng trong huấn luyện trực tuyến bộ điều khiển. Bởi lẽ, kỹ thuật này đòi hỏi vừa lấy mẫu dữ liệu vừa huấn luyện và điều khiển, cho nên việc cải thiện thời gian trong từng công đoạn, nhất là thời gian huấn luyện có ý nghĩa quyết định đến chất lượng của bộ điều khiển.

4 KẾT LUẬN

Trong bài báo này, giải thuật Quasi-Newton đã được sử dụng để huấn luyện trực tuyến mạng nơ-ron RBF trong bộ điều khiển trượt thích nghi robot di động đa hướng. Kết quả mô phỏng với

MATLAB/SIMULINK cho thấy hiệu quả của giải thuật này với các chỉ tiêu chất lượng cũng như các tiêu chí RMS và MSE vượt trội hơn so với giải thuật Gradient Descent. Điều đó chứng tỏ giải thuật đề xuất phù hợp để điều khiển robot di động đa hướng cũng như trong điều khiển hệ phi tuyến.

TÀI LIỆU THAM KHẢO

- Likas, A. and Stafylopatis, A., 2000. Training the Random Neural Network using Quasi-Newton methods. *European Journal of Operational Research*. 126(2): 331–339.
- Sharma, B. and K. Venugopalan, 2014. Comparison of Neural Network Training Functions for Hematoma Classification in Brain CT Images. *IOSR Journal of Computer Engineering (IOSR-JCE)*. 16(1): 31-35.
- Mukherjee, I. and Routroy, S., 2012. 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 Systems with Applications* 39(3): 2397–2407.
- Liu, J., 2013. Radial Basis Function (RBF) Neural Network Control for Mechanical Systems. Springer, 374 pages.
- Watanabe, K., 1998. Control of an omnidirectional Mobile Robot. Second International Conference on Knowledge-Based Intelligent Electronic Systems: 51 - 60.
- Sadeghi, M., Pashaie, M., and Jafarian, A., 2016. RBF Neural Networks Based on BFGS Optimization Method for Solving Integral Equations. *Advances in Applied Mathematical Biosciences*. 7(1): 1-22.
- Relhan, N., Jain, M., 2012. Analysis of Optimization Techniques for Feed Forward Neural Networks Based Image Compression. *International Journal of Computer Science and Information Technologies*. 3(2): 3291-3294.
- Mahmoud, O., Anwar, F., Jimoh, M. and Salami, E., 2007. Learning Algorithm Effect on Multilayer Feed Forward Artificial Neural Network Performance in Image Coding. *Journal of Engineering Science and Technology*. 2(2): 188 – 199.
- Djebrani, S.A., Benali, A. and Abdessemed, F., 2012. Modelling and control of an omnidirectional mobile manipulator. *Int. J. Appl. Math. Comput. Sci.* 22(3): 601–616.
- Burney, S.M.A., Jilani, T.A. and Ardil, C., 2007. A Comparison of First and Second Order Training Algorithms for Artificial Neural Networks. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*. 1(1): 145 – 151.
- Yang, Y.C. and Cheng, C.C., 2013. Robust adaptive trajectory control for an Omnidirectional vehicle with parametric uncertainty. *Transactions of the Canadian Society for Mechanical Engineering*. 37(3): 405 – 413.