ADAPTIVE SLIDING MODE CONTROL WITH RBF NEURAL NETWORKS FOR OMNI-DIRECTIONAL MOBILE ROBOT

ĐIỀU KHIỂN TRƯỢT THÍCH NGHI VỚI MẠNG NƠ-RON RBF CHO ROBOT DI ĐỘNG ĐA HƯỚNG

Thanh-Tung Pham[1], Chi-Ngon Nguyen[2], Ngo-Phong Nguyen[3], Huong Dong Van[4], Kieu-Mai Le Thi[1], Hoang-Tam Vo[1]

[1]Vinh Long University of Technology Education, Vietnam, [2]Can Tho University, Vietnam
[3] Ulsan National Institute of Science and Technology (UNIST), Korea
[4]Ho Chi Minh City University of Transportation, Vietnam

Received 30/3/2018, Peer reviewed 4/6/2018, Accepted for publication 03/7/2018

ABSTRACT

In this paper, we consider the adaptive sliding mode control with radial basis function neural networks for the Omni-directional mobile robot. This is a holonomic robot that can operate easily in small and narrow spaces, due to the ability of flexible rotational and translational moving, simultaneously and independently. This robot is a MIMO nonlinear system. We design the sliding mode control (SMC) to ensure the trajectory tracking problem for a mobile robot. Therein, the radial basis function (RBF) neural networks are trained and used to approximate the adaptive SMC control law. In addition, the parameters of the neural networks are updated during the operation by using the gradient descent algorithm. Furthermore, we show the asymptotical convergence of the system state with the proposed control strategy. Finally, the simulation is conducted to verify the effectiveness of the proposed control system under disturbances and system uncertainties. These results demonstrate that the proposed algorithm is feasible to control the robot as well as control the nonlinear systems.

***Keywords:** sliding mode control; adaptive law; radial basis function neural networks; Omni-directional mobile robot; Gradient Descent algorithm.*

TÓM TẮT

Bài báo trình bày phương pháp thiết kế bộ điều khiển trượt thích nghi sử dụng mạng nơ-ron RBF (Radial Basis Function) để điều khiển bám quỹ đạo 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. Robot này là hệ phi tuyến MIMO. Bộ điều khiển trượt được thiết kế để đảm bảo quỹ đạo thực tế của robot bám theo quỹ đạo cho trước. 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 và đóng vai trò như một bộ điều khiển thích nghi. Các trọng số của mạng được cập nhật trực tuyến bằng giải thuật Gradient Descent dựa trên các tín hiệu hồi tiếp ở ngõ ra. Kết quả mô phỏng trên MATLAB/ SIMULINK cho thấy rằng giải thuật đề xuất đáp ứng được mục tiêu điều khiển ngay cả khi có nhiễu tác động lên hệ thống hoặc thông số của mô hình thay đổi theo thời gian với sai số bám tiến về 0, thời gian quá độ khoảng 0.3(s). 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.

***Từ khóa:** điều khiển trượt; thích nghi; mạng nơ-ron RBF; robot di động đa hướng; giải thuật Gradient Descent.*

1. INTRODUCTION

The class of SMC is considered as an efficient algorithm to ensure the stability of the system under disturbances or system uncertainties in the different nonlinear dynamic model [1-3]. However, it requires explicit information about the dynamic model, which is not easy to meet in the practical control situation. Although the SMC guarantees the robustness of the system against the disturbances and the uncertainties, the chattering phenomenon always occurs, which can degrade the control performance of the system.

The Omni-directional provides an outstanding feature for moving in the narrow space. This mobile robot is considered a very complex dynamic system. Therefore, various control method had been proposed to deal with the trajectory tracking control of Omni-directional mobile robot, including fuzzy logic control [4], feedback linearization control [5], and adaptive control [6-7].

In this paper, we consider the adaptive sliding mode control with radial basis function neural network for an Omni-directional mobile robot. We design the sliding mode control to ensure the trajectory tracking problem for a mobile robot. Therein, the radial basis function neural network, which is considered as an adaptive law, is developed to estimate the nonlinear function in the proposed SMC law. In addition, the parameters of the neural network are updated during the operation by using the Gradient descent algorithm. Furthermore, we show the asymptotical convergence of the system state with the proposed control strategy. Finally, the simulation is conducted to verify the effectiveness of the proposed control architecture under disturbances and system uncertainties.

2. ADAPTIVE SLIDING MODE CONTROL WITH RADIAL BASIS FUNCTION NEURAL NETWORK FOR OMNI-DIRECTIONAL MOBILE ROBOT

2.1. Dynamic model of the Omni-directional mobile robot

Let us consider the mobile robot as a rigid object moving on the workspace. Let assume that the absolute coordinate system $O_w - X_wY_w$ is fixed on the plane and the moving coordinate system $O_m - X_mY_m$ is fixed in the center of the mobile robot system as given in Figure. 1.

The center gravity of the mobile robot is defined by the following vector $S_w = \begin{bmatrix} x_w & y_w \end{bmatrix}^T$. Then, we have:

$$\mathcal{M}\ddot{S}_w = F_w \tag{1}$$

where $F_w = \begin{bmatrix} F_x & F_y \end{bmatrix}^T$ is the force vector in the absolute coordinate system which is applied in the center gravity of the mobile robot and $\mathcal{M}$ is the symmetric positive-definite matrix with the mass M, $\mathcal{M} = diag\left(\left[M, M\right]\right)$.

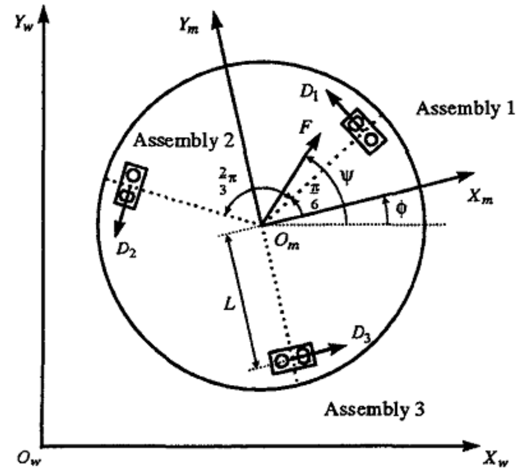

***Figure. 1**. Robot coordinate framework [8]*

Let ϕ represent the angle between X_w- và X_m- , which is the rotational angle of the moving coordinate system with respect to the absolute coordinate system. Hence, we can obtain the coordinate transformation matrix from the absolute coordinate system to the moving coordinate system as:

$${}^{w}R_m = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{2}$$

Then, we have:

$$\dot{S}_w = {}^wR_m\dot{s}_m \quad (3)$$

$$F_w = {}^wR_m f_m \quad (4)$$

where $s_m = \begin{bmatrix} x_m & y_m \end{bmatrix}^T$ denote the position vector of the center gravity and $f_m = \begin{bmatrix} f_x & f_y \end{bmatrix}^T$ is the force vector applied to the center gravity in the moving coordinate system.

Thus, transforming (1) to the moving coordinate system, we have:

$$\mathcal{M}\left({}^wR_m^T\,{}^w\dot{R}_m\dot{s}_m + \ddot{s}_m\right) = f_m \quad (5)$$

Then, the dynamic model of the mobile robot can be described as:

$$M\left(\ddot{x}_m - \dot{y}_m\dot{\phi}\right) = f_x \quad (6)$$

$$M\left(\ddot{y}_m + \dot{x}_m\dot{\phi}\right) = f_y \quad (7)$$

$$I_v\ddot{\phi} = M_I \quad (8)$$

where I_v and M_I are the moment of inertia and the moment around the center gravity of the robot, respectively, and f_x, f_y, M_I are given by:

$$f_x = -\frac{1}{2}D_1 - \frac{1}{2}D_2 + D_3 \quad (9)$$

$$f_y = \frac{\sqrt{3}}{2}D_1 - \frac{\sqrt{3}}{2}D_2 \quad (10)$$

$$M_I = \left(D_1 + D_2 + D_3\right)L \quad (11)$$

Let us consider the driving system property for each assembly is assumed to be given by:

$$I_w\dot{w}_i + cw_i = ku_i - rD_i \quad (\mathrm{i} = 1,2,3) \quad (12)$$

where L is the distance between any assembly and the center gravity of the robot; c is the viscous friction for the wheel; D_i is the driving force for each assembly; r is the radius of the wheel; I_v is the moment inertia of the wheel around the driving shaft; w_i is the rotational rate of the wheel; k is the driving gain factor; u_i is the driving input torque.

In addition, the geometrical relationship among variables $\dot{\phi}, \dot{x}_m, \dot{y}_m, w_i, \ldots$ that is the inverse kinematics are given by:

$$rw_1 = -\frac{1}{2}\dot{x}_m + \frac{\sqrt{3}}{2}\dot{y}_m + L\dot{\phi} \quad (13)$$

$$rw_2 = -\frac{1}{2}\dot{x}_m - \frac{\sqrt{3}}{2}\dot{y}_m + L\dot{\phi} \quad (14)$$

$$rw_3 = \dot{x}_m + L\dot{\phi} \quad (15)$$

Hence, by using from (6) to (15), we have:

$$\ddot{x}_m = a_1\dot{x}_m + a_2\dot{y}_m\dot{\phi} - b_1\left(u_1 + u_2 - 2u_3\right) \quad (16)$$

$$\ddot{y}_m = a_1\dot{y}_m + a_2\dot{x}_m\dot{\phi} + \sqrt{3}b_1\left(u_1 - u_2\right) \quad (17)$$

$$\ddot{\phi} = a_3\dot{\phi} + b_2\left(u_1 + u_2 + u_3\right) \quad (18)$$

where:

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

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

$$b_2 = \frac{krL}{(3I_w + I_vr^2)}$$

Finally, from (16)-(18), we can obtain:

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\phi}_w \end{bmatrix} = \begin{bmatrix} a_1 & -a_2\phi_d & 0 \\ a_2\phi_d & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix}\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi}_w \end{bmatrix} + \begin{bmatrix} b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi \\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi \\ b_2 & b_2 & b_2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} D_{fx} \\ D_{fy} \\ D_{f\phi} \end{bmatrix} \quad (19)$$

$$= A_W\beta + B_WU + D_f$$

where $D_f = \begin{bmatrix} D_{fx} & D_{fy} & D_{f\phi} \end{bmatrix}^T$ represents the disturbances, and

$$a_2 = 1 - a_2' = \frac{3I_w}{(3I_w + 2Mr^2)}$$

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

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

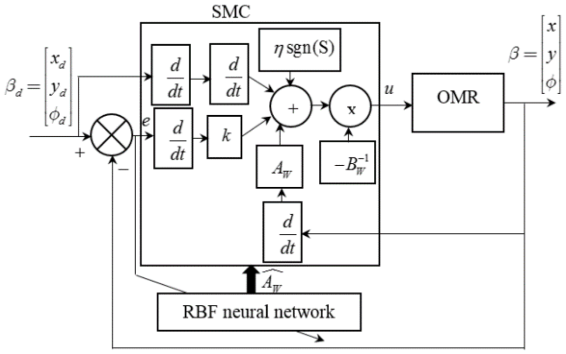

Figure. 2. *Block diagram of the proposed system*

2.2. Adaptive sliding mode control with radial basis function neural network

The overall block diagram of the adaptive sliding mode control with radial basis function neural network is given in Figure 2.

Let us define the following error vector:

$$e = \beta - \beta_d = \begin{bmatrix} e_x \\ e_y \\ e_\phi \end{bmatrix} \tag{20}$$

where $\beta = \begin{bmatrix} x_w & y_w & \phi_w \end{bmatrix}^T$ and $\beta_d = \begin{bmatrix} x_d & y_d & \phi_d \end{bmatrix}^T$.

Next, the sliding surface is given by:

$$S = \begin{bmatrix} (\dot{x}_w - \dot{x}_d) + k_x (x_w - x_d) \\ (\dot{y}_w - \dot{y}_d) + k_y (y_w - y_d) \\ (\dot{\phi}_w - \dot{\phi}_d) + k_\phi (\phi_w - \phi_d) \end{bmatrix} = \begin{bmatrix} \dot{e}_x + k_x e_x \\ \dot{e}_y + k_y e_y \\ \dot{e}_\phi + k_\phi e_\phi \end{bmatrix} = \begin{bmatrix} S_x \\ S_y \\ S_\phi \end{bmatrix} \tag{21}$$

where $k = diag\left(\begin{bmatrix} k_x & k_y & k_\phi \end{bmatrix}\right)$ with positive con-stant parameters. Hence, (21) can be rewritten in a form $S = \begin{bmatrix} S_x & S_y & S_\phi \end{bmatrix}^T$, $d(t)$.

Finally, the proposed control law is as (22):

$$U = -B_W^{-1} \ A_W \dot{\beta} + \ \ddot{\beta}_d + k\dot{e} \ + \eta \operatorname{sgn} \ S \tag{22}$$

Where $\eta = diag\left(\left[\eta_x, \eta_y, \eta_\phi\right]\right)$, and the approximation error is given by:

$$\Gamma = \Gamma - \hat{\Gamma}$$

with $\Gamma = A_W$.

In this paper, we develop the radial basis function neural network to approximate A_W. The general block diagram of the radial basis function neural network is given in Figure. 3.

In Figure. 3, h_j is given by (23):

$$h_j = \exp\left(-\frac{\left\| x - c_j \right\|^2}{2b_j^2} \right) \tag{23}$$

The radial basis function neural network which is developed to approximate the function f_1 of the matrix A_W is illustrated in Figure. 4.

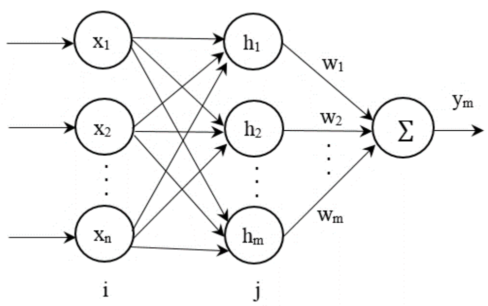

Figure. 3. *Block diagram of the radial basis function neural network*

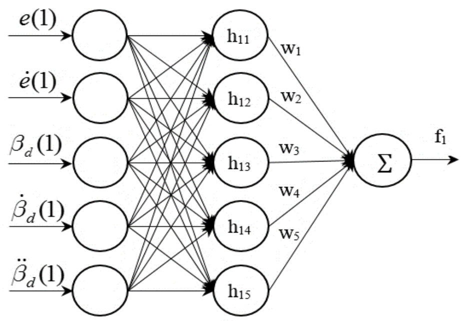

Figure. 4. *Block diagram of the proposed neural network*

where

$$x_1 = \begin{bmatrix} e(1) & \dot{e}(1) & \beta_d(1) & \dot{\beta}_d(1) & \ddot{\beta}_d(1) \end{bmatrix}$$

$$h_1 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} & h_{15} \end{bmatrix}$$

$$W = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 \end{bmatrix}$$

The output of the proposed neural network is given by:

$$f_1 = W^T h_1 \tag{24}$$

By applying the same technique, we can obtain f_2 and f_3.

For f_2:

$$x_2 = \begin{bmatrix} e(2) & \dot{e}(2) & \beta_d(2) & \dot{\beta}_d(2) & \ddot{\beta}_d(2) \end{bmatrix}$$

$$h_2 = \begin{bmatrix} h_{21} & h_{22} & h_{23} & h_{24} & h_{25} \end{bmatrix}$$

$$V = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix}$$

The output f_2 can be obtained as:

$$f_2 = V^T h_2 \tag{25}$$

For f_3:

$$x_3 = \begin{bmatrix} e(3) & \dot{e}(3) & \beta_d(3) & \dot{\beta}_d(3) & \ddot{\beta}_d(3) \end{bmatrix}$$

$$h_3 = \begin{bmatrix} h_{31} & h_{32} & h_{33} & h_{34} & h_{35} \end{bmatrix}$$

$$Z = \begin{bmatrix} z_1 & z_2 & z_3 & z_4 & z_5 \end{bmatrix}$$

The output f_3 can be obtained as:

$$f_3 = Z^T h_3 \tag{26}$$

Hence, the approximation matrix A_W can be rewritten in the following form:

$$\hat{\Gamma} = \begin{bmatrix} f_1 & -f_2\phi_d & 0 \\ f_2\phi_d & f_1 & 0 \\ 0 & 0 & f_3 \end{bmatrix} = \begin{bmatrix} W^T h_1(x) & -V^T \phi_d h_2(x) & 0 \\ V^T h_2(x) & W^T h_1(x) & 0 \\ 0 & 0 & Z^T h_3(x) \end{bmatrix} \tag{27}$$

Now, let us verify the stability of the system with the proposed control strategy. From (21), we have:

$$S = \dot{e} + ke \tag{28}$$

where $c > 0$. Then, taking the derivative of S, we have:

$$\begin{aligned} \dot{S} &= \ddot{e} + k\dot{e} = \ddot{\beta}_d - \ddot{\beta} + k\dot{e} \\ &= \ddot{\beta}_d - \left(A_W + \Delta A_W\right)\begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\phi}_w \end{bmatrix} - B_W(\phi)\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - d(t) \\ &= -\eta \operatorname{sgn}(S) - d(t) \end{aligned} \tag{29}$$

Hence, if $\eta \geq D$, we can conclude:

$$S\dot{S} = -\eta|S| - S.d(t) \leq 0$$

From (22) and (29), we have:

$$\begin{aligned} \dot{S} &= \ddot{\beta}_d - A_W\dot{\beta} - B_W U - d\ \ t\ \ + k\dot{e} \\ &= \ddot{\beta}_d - A_W\dot{\beta} - \ \ -A_W\dot{\beta} + \ddot{\beta}_d + k\dot{e} + \eta \operatorname{sgn}\ \ S \\ &\quad - d\ \ t\ \ + k\dot{e} \\ &= -A_W\dot{\beta} + A_W\dot{\beta} - \eta \operatorname{sgn}\ \ S\ \ - d\ \ t \\ &= -A_W\dot{\beta} - d\ \ t\ \ - \eta \operatorname{sgn}\ \ S \end{aligned} \tag{30}$$

where $\Gamma = \Gamma - \hat{\Gamma}$, $\Gamma = A_W$, $\hat{\Gamma} = A_W$

Let us introduce the following Lyapunov function candidate:

$$V = \frac{1}{2} S^T S > 0$$

Taking the derivative of V, we have:

$$\begin{aligned} \dot{V} &= S^T\ \ -A_W\,\dot{\beta} - \eta \operatorname{sgn}\ \ S \\ &= S^T\ \ -\tilde{\Gamma}\dot{\beta} - \eta \operatorname{sgn}\ \ S \\ &= -S^T\ \ \tilde{\Gamma}\dot{\beta} + \eta \operatorname{sgn}\ \ S\ \ \ \leq 0 \end{aligned}$$

where η is the symmetric positive-definite matrix. Therefore, we can conclude that $S(t) \to 0$ as $t \to \infty$. Therefore, $e(t), \dot{e}(t) \to 0$ as $t \to \infty$.

2.3. Simulation results

The parameters of the system and controller are respectively given in ***Table 1, 2*** and ***3*** [8].

Table 1. *The parameters of the mobile robot*

Notation	Meaning	Value	Unit
I_v	Moment inertia of the mobile robot	11.25	kgm^2
M	Mass of the robot	9.4	kg
L	Distance between any assembly and the center gravity of the robot	0.178	m

k	Driving gain factor	0.448	
c	Viscous friction factor of the wheel	0.1889	kgm^2/s
I_ω	Moment inertia of the wheel	0.02108	kgm^2
r	Radius of the wheel	0.0245	m

Table 2. *The parameters of the adaptive sliding mode control*

Meaning	Notation	Value
Sliding paramters	$\begin{bmatrix} \eta_x & 0 & 0 \\ 0 & \eta_y & 0 \\ 0 & 0 & \eta_\phi \end{bmatrix}$	$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$
	$\begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\phi \end{bmatrix}$	$\begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 58 \end{bmatrix}$

Table 3. *The parameters of the radial basis function neural network*

Notation	Meaning	Value
α	Momentum factor	0.05
μ	Learning factor	$\begin{bmatrix} 0.5 & 0.5 & 0.5 \end{bmatrix}^T$
c_{ij}	Center vector for the neural network	$\begin{bmatrix} -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \\ -1 & -0.5 & 0 & 0.5 & 1 \end{bmatrix}$
b_j	Certain threshold of the neural network	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$
$\begin{bmatrix} W \\ V \\ Z \end{bmatrix}$	Parameter vector	$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 & 0.5 & 0.5 \end{bmatrix}$
i	Total inputs	5
j	Total neurons in hidden layers	5

The desired trajectory is given by:

$$x_d = 0.3\cos(2\pi t)\ (m)$$
$$y_d = 0.3\sin(2\pi t)\ (m) \quad (31)$$
$$\phi_d = -0.2\pi t\ (rad/s)$$

The performance of the ASMC-RBF is given in Figures. 5-7. Therein, the ASMC-RBF provides a superior performance in compared with the traditional SMC in terms of settling time, steady-state error and overshoot, as given in Table 4.

Table 4*. Control performance*

	ASMC-RBF	RBF
Steady-state error	$6.25\ .10^{-5}$	$6.66\ .10^{-5}$
Settling time	0.3(s)	0.3(s)
Overshoot	$0.0399.10^{-4}$ (%)	$0.0401.10^{-4}$ (%)

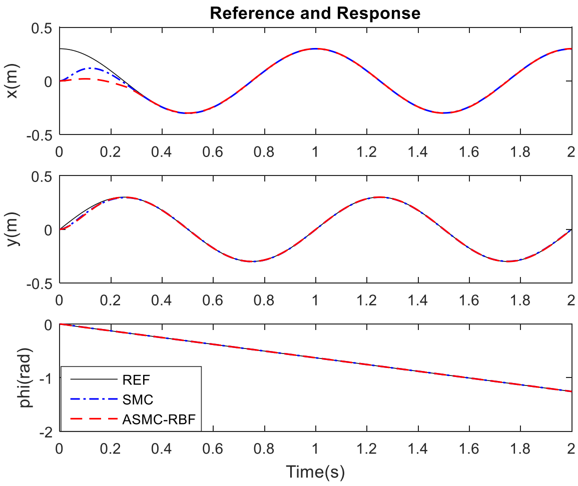

Figure. 5. *Control state performance*

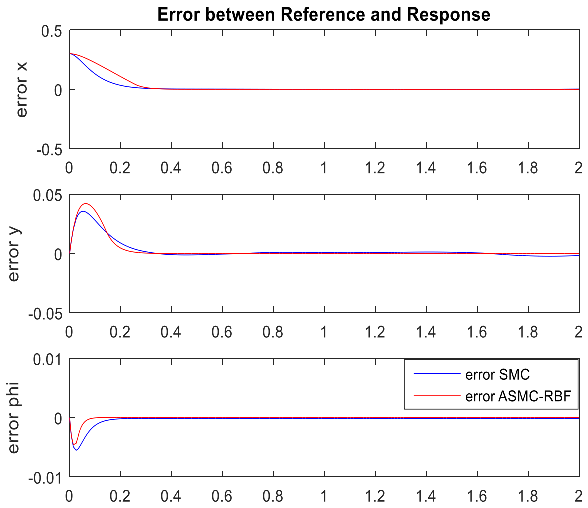

Figure. 6. *Error response*

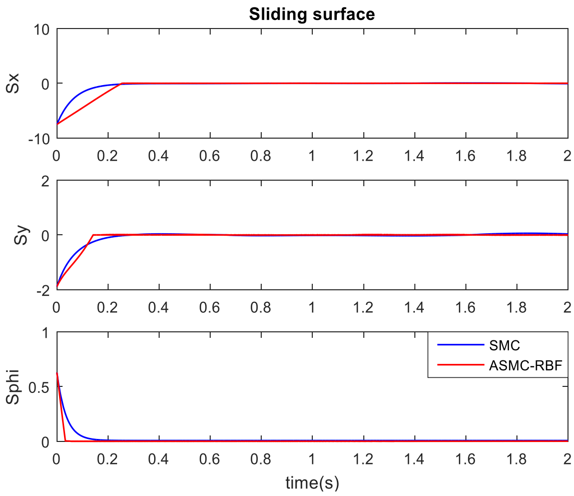

Figure. 7. *Sliding surface*

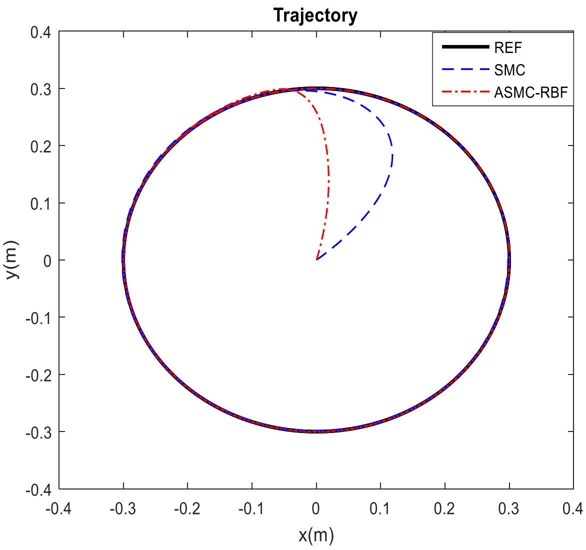

Figure. 8. *Trajectory tracking performance*

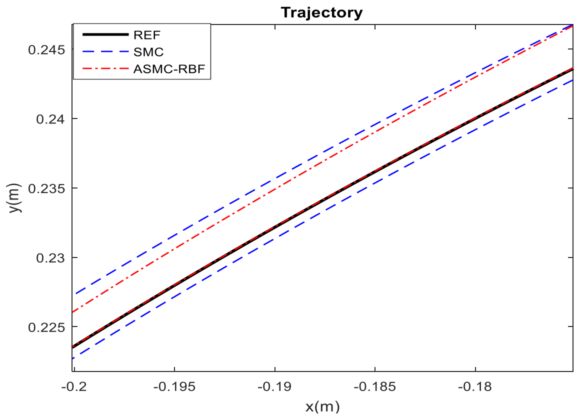

Figure. 9. *Zoom in trajectory tracking performance*

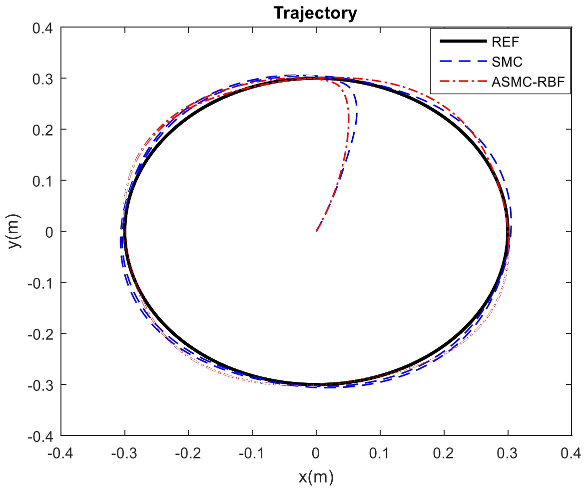

Figure. 10. *Trajectory tracking performance under measurement noise*

The tracking control performance is given in Figure. 8, which verify the control performance of the proposed control strategy and the traditional SMC. Once again, it confirms the superior control performance of the ASMC-RBF in compared with SMC, see Figure. 9 for the zoom version.

In addition, we also consider the performance of the ASMC-RBF under the measurement noise 0.001(W). The system response is given in Figure. 10 which shows that stability is still ensured by the proposed control strategy.

3. CONCLUSION

We addressed the adaptive sliding mode control with radial basis function neural network for Omni-directional mobile robot.

We develop the SMC to guarantee the trajectory tracking problem for mobile robot. Therein, the proposed control strategy is featured by the radial basis function neural network which is developed to estimate the nonlinear function in the proposed SMC law. In addition, the parameters of the neural network are updated during the operation by using the Gradient descent algorithm. Furthermore, we show the asymptotically convergence of the system state with the proposed control strategy. Numerical simulation is conducted to verify the effectiveness of the proposed control architecture under disturbances and system uncertainties.

REFERENCES

[1] Hoang Dung Nguyen (2010), *Adaptive sliding mode control using radial basic function neural network*, Can Tho University Journal of Science 2010:15a, pp. 263 - 272.

[2] Ali J. Koshkouei, Alan S. I. Zinober, and Keith J. Burnham (2004), *Adaptive sliding mode backstepping control of nonlinear systems with unmatched uncertainty,* Asian Journal of Control, Vol. 6, No. 4, pp. 447-453.

[3] Ghania Debbache and Noureddine Goléo (2012), *Neural network based adaptive sliding mode control of uncertain nonlinear systems*, Journal of Systems Engineering and Electronics, Vol.23, No.1, pp. 119 - 128.

[4] Sangdae Kim, Changho Hyun, Youngwan Cho, Seungwoo Kim: *Tracking Control of 3-Wheels Omni-directional Mobile RobotUsing Fuzzy Azimuth Estimator*, Proceedings of the 10th WSEAS Int. Conference on Robotics, Control and Manufacturing Technology, pp. 47 – 51.

[5] Yong Liu a, J. Jim Zhu a, Robert L. Williams IIb, Jianhua Wu (2008): *Omni-directional mobile robot controller based on trajectory linearization*, Robotics and Autonomous Systems 56, pp. 461–479.

[6] Ya-Chao Yang and Chi-Cheng Cheng (2013), *Robust adaptive trajectory control for an omnidirectional vehicle with parametric uncertainty*, Transactions of the Canadian Society for Mechanical Engineering, Vol. 37, No. 3, pp. 405 - 413.

[7] A. Salam Al-Ammri Iman Ahmed (2010): *Control of Omni-directional Mobile Robot Motion*, Al-Khwarizmi Engineering Journal, Vol. 6, No. 4, pp 1 – 9.

[8] Keigo Watanabe (1998), *Control of an omnidirectional Mobile Robot*, Second International Conference on Knowledge-Based IntelligentElectronic Systems, 21-23 April 1998, pp. 51 - 60.

Corresponding Author:
Pham Thanh Tung
Vinh Long University of Technology Education
Email: tungpt@vlute.edu.vn