Omnidirectional Mobile Robot Trajectory Tracking Control with Diversity of Inputs

Thanh Tung Pham and Minh Thanh Le
Vinh Long University of Technology Education, Vietnam
Email: tungpt@vlute.edu.vn, thanhlm@vlute.edu.vn

Chi-Ngon Nguyen (*)
Can Tho University, Vietnam
Email: ncngon@ctu.edu.vn

Abstract—**This study aims to evaluate the control quality criteria for uncertain systems with diversity of reference signals. For being easy, the control algorithm is developed and tested on a model of omnidirectional mobile robot to evaluate the performance of the proposed method. The omnidirectional mobile robot is a holonomic robot that has been widely used for surveillance, inspection and transportation tasks. The radial basis function (RBF) neural network - based adaptive sliding mode control (SMC) for each input is compared with a classical SMC on the robot model. The RBF neural networks are used to estimate the nonlinear functions in the SMC law. By online training mechanism, the SMC law can adapt to the changes of control conditions. Simulations in MATLAB/Simulink indicate that the system responses are stable without steady-state error, and the overshoots archive 0.4 (%). Results illustrate that the RBF neural networks–based adaptive SMC control is stable with diversity of inputs.**

Index Terms—**Omnidirectional mobile robot, sliding mode control, RBF neural networks, uncertain systems**

I. INTRODUCTION

Mobile robots have been widely used for technical, industrial, and educational applications [1]. One of the main requirements of an autonomous mobile robot is its ability to move through the operational space, avoiding obstacles and finding its way to the next location, in order to perform its task, capabilities known as localization and navigation. In order to move in tight areas and to avoid obstacles mobile robots should have good mobility and maneuverability. These capabilities mainly depend on the controller design [2]. In recent years, due to mobile robots' practical importance, the problem of trajectory tracking control has attached a great deal of attention from the nonlinear control community, for instance, in reference [3] used the trajectory linearization control (TLC) method based on a nonlinear robot dynamic model to control the robot. In reference [4], a direct adaptive controller based on improved RBF neural network is proposed for an Omni-directional mobile robot (OMR). The simulation results demonstrate the feasibility and validity of proposed scheme. A fuzzy PID controller was designed in reference [5]. The result shown that the mobile robot with the fuzzy PID controller can track the desired references by 3 seconds. In reference [6], a nonlinear controller is designed for a wheeled mobile robot with three Omni-directional wheels. In controller design, a computed torque control law is applied to counterbalance the nonlinear terms of the system. Then a sliding mode controller is developed for trajectory tracking. The effectiveness of the proposed control strategy is demonstrated by computer simulations. The problem is that when changing the robot's dynamic behaviors, those fixed controllers are not flexible enough to the control conditions.

In this paper, firstly, an adaptive sliding mode control (SMC) law based on radial basis function (RBF) neural network will be designed to control uncertain systems so that all the signals of the closed-loop system are uniformly ultimately bonded and the trajectory tracking errors converge to small regions in finite time. For being easy to understand, the Omnidirectional mobile robot is selected for testing the effectiveness of the proposed algorithm. Secondly, this controller will be verified the robust capabilities of it's with difference inputs.

The rest of this paper is organized as follow: Section II presents the robot dynamic model; the RBF neural network -based adaptive sliding mode control is given in Section III; Section IV shows the simulation results in MATLAB/Simulink with diversity inputs; and the conclusion is given in Section V.

II. DYNAMIC MODEL OF THE ROBOT

Assuming that the mobile robot only moves on its work-space. And the absolute coordinate O_w - X_wY_w is fixed on the plane, and the motion coordinate $O_m - X_mY_m$ is also fixed on the center of gravity, as presented in Fig. 1 [7].

Manuscript received November 16, 2020; revised March 22, 2021.
* Corresponding author: Chi-Ngon Nguyen, Can Tho University, 94000 Can Tho, Vietnam, Email: ncngon@ctu.edu.vn

doi: 10.18178/ijmerr.10.11.639-644

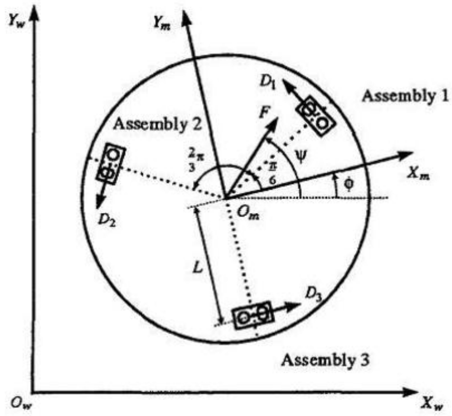

Figure 1. The model of robot [7].

The state-space model can be described by (1) [7]:

$$\begin{bmatrix}\ddot{x}_w\\ \ddot{y}_w\\ \ddot{\phi}\end{bmatrix}=\begin{bmatrix}a_1 & -a_2\phi & 0\\ a_2\phi & a_1 & 0\\ 0 & 0 & a_3\end{bmatrix}\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\phi}\end{bmatrix}+\begin{bmatrix}b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi\\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi\\ b_2 & b_2 & b_2\end{bmatrix}\begin{bmatrix}u_x\\ u_y\\ u_\phi\end{bmatrix}+\begin{bmatrix}D_{fx}\\ D_{fy}\\ D_{f\phi}\end{bmatrix}=\boldsymbol{A}_w\boldsymbol{x}+\boldsymbol{B}_w\boldsymbol{u}+\boldsymbol{D}_f \tag{1}$$

$$y(t)=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}_w\\ \dot{y}_w\\ \dot{\phi}_w\end{bmatrix}=\boldsymbol{C}_w\boldsymbol{x} \tag{2}$$

where the state variables are $\boldsymbol{x}=\left[\dot{x}_w,\dot{y}_w,\dot{\phi}_w\right]^T$, the outputs are $\boldsymbol{y}=\left[\dot{x}_w,\dot{y}_w\right]^T$, manipulated variables are denoted by $\boldsymbol{u}=\left[u_x,u_y,u_\phi\right]^T$, and $D_f=\left[D_{fx},D_{fy},D_{f\phi}\right]^T$ are unknown system disturbances. $\boldsymbol{A}_w$ and $\boldsymbol{B}_w$ are calculated based on robot's parameters as

$$\boldsymbol{A}_w=\begin{bmatrix}a_1 & -a_2\dot{\phi} & 0\\ a_2\dot{\phi} & a_1 & 0\\ 0 & 0 & a_3\end{bmatrix};\ \boldsymbol{B}_w=\begin{bmatrix}b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi\\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi\\ b_2 & b_2 & b_2\end{bmatrix},$$

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

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

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

where M is the mass of the robot; L is the distance between any assembly and the center of gravity of the robot; c is the viscous friction factor for the wheel; r is the radius of the wheel; I_w is the moment of inertia of the wheel around the driving shaft; k is the driving gain factor; and I_v is the moment of inertia for the robot.

III. Adaptive Sliding Mode RBF Neural Networks Control

A. Sliding Mode Controller Design

Sliding mode control (SMC) was first proposed and elaborated in the early 1950s by Emelyanov *et.al* [8]. During the last decades, significant interest on SMC has been generated in the control research community. In this section, the SMC will be designed to keep the robot's actual trajectories follows the references and tracking errors converge to zero in a finite time.

We assume that the notation x_d, y_d and ϕ_d are the reference signals; the notation x_w, y_w and ϕ_w are the reality outputs of the robot. The SMC is designed to control the robot so that $x_w\rightarrow x_d$, $y_w\rightarrow y_d$ and $\phi_w\rightarrow\phi_d$ in a finite time.

In order to design the control signal for x_w, (1) can be written with x_w as (3):

$$\ddot{x}_w=a_1\dot{x}_w-a_2\phi\dot{y}_w+b_1\gamma_1u_x+b_1\gamma_1u_y+2b_1\cos\phi u_\phi+D_{fx} \tag{3}$$

The error between x_d and x_w is defined as (4):

$$e_x=x_w-x_d \tag{4}$$

The sliding surface as (5):

$$S_x=\dot{e}_x+k_xe_x=\left(\dot{x}_w-\dot{x}_d\right)+k_x\left(x_w-x_d\right) \tag{5}$$

$$\Rightarrow\dot{S}_x=\ddot{e}_x+k_x\dot{e}_x=\left(\ddot{x}_w-\ddot{x}_d\right)+k_x\left(\dot{x}_w-\dot{x}_d\right) \tag{6}$$

where x_d is the reference input and k_x is performance parameter which guarantees the stability of the system ($k_x>0$). On sliding surface:

$$S_x=0\rightarrow\dot{S}_x=0 \tag{7}$$

Equivalent control can be found by substituting (3) into (6) when $\dot{S}_x=0$ as follow

$$\begin{aligned}\dot{S}_x&=\left(\ddot{x}_w-\ddot{x}_d\right)+k_x\left(\dot{x}_w-\dot{x}_d\right)\\ &=a_1\dot{x}_w-a_2\phi\dot{y}_w+b_1\gamma_1u_x+b_1\gamma_1u_y\\ &\quad+2b_1\cos\phi u_\phi+D_{fx}-\ddot{x}_d+k_x\left(\dot{x}_w-\dot{x}_d\right)=0\end{aligned} \tag{8}$$

$$u_{eqx}=u_x=\frac{1}{b_1\gamma_1}\begin{pmatrix}-a_1\dot{x}_w+a_2\phi\dot{y}_w-b_1\gamma_2u_y\\ -2b_1\cos\phi u_\phi-D_{fx}\\ +\ddot{x}_d-k_x\left(\dot{x}_w-\dot{x}_d\right)\end{pmatrix} \tag{9}$$

Switching control is chosen as (10):

$$u_{swx}=\eta_x sign\left(S_x\right) \tag{10}$$

Therefore, the final SMC law for x_w signal as (11):

$$u_{SMCx}=-\frac{1}{b_1\gamma_1}\begin{pmatrix}a_1\dot{x}_w-a_2\phi\dot{y}_w+b_1\gamma_2u_y\\ +2b_1\cos\phi u_\phi-\ddot{x}_d+k_x\dot{e}_x\\ +\eta_x sign\left(S_x\right)+D_{fx}\end{pmatrix} \tag{11}$$

Similar as above, we have the SMC law for y_w signal as (12) and the SMC law for ϕ_w signal as (13):

$$u_{SMCy} = -\frac{1}{b_1\gamma_4}\begin{pmatrix} a_1\dot{y}_w + a_2\phi\dot{x}_w + b_1\gamma_3 u_x \\ +2b_1\sin\phi u_\phi - \ddot{y}_d + k_y\dot{e}_y \\ +\eta_y sign(S_y) + D_{fy} \end{pmatrix} \quad (12)$$

$$u_{SMC\phi} = -\frac{1}{b_2}\begin{pmatrix} a_3\dot{\phi} + b_2 u_x + b_2 u_y \\ -\ddot{\phi}_d + k_\phi \dot{e}_\phi + \eta_\phi sign(S_\phi) + D_{f\phi} \end{pmatrix} \quad (13)$$

Finally, the SMC law for the robot is presented as (14):

$$\boldsymbol{U}_{SMC} = -\boldsymbol{B}_W^{-1}\left\{\boldsymbol{A}_W\dot{\boldsymbol{\beta}} - \ddot{\boldsymbol{\beta}}_d + \boldsymbol{k}\dot{\boldsymbol{e}} + \boldsymbol{\eta} sign(\boldsymbol{S}) + \boldsymbol{D}_f\right\} \quad (14)$$

where $\boldsymbol{\eta} = diag[\eta_x, \eta_y, \eta_\phi]$ are symmetric positive definite; $\boldsymbol{e} = \boldsymbol{\beta} - \boldsymbol{\beta}_d$ are the tracking errors with $\boldsymbol{\beta} = [x_w, y_w, \phi_w]^T$ are the actual trajectories and $\boldsymbol{\beta} = [x_d, y_d, \phi_d]^T$ are the references, $S = [S_x, S_y, S_\phi]^T$ are sliding mode surface functions and $\boldsymbol{k} = diag[k_x, k_y, k_\phi]$ are designed positive constants.

As defined in (1), the determinant of $\boldsymbol{B}_w$ as (15):

$$\det(\boldsymbol{B}_W) = 6\sqrt{3}b_1^2 b_2 \neq 0 \quad (15)$$

Because of determinant of the matrix $\boldsymbol{B}_w$ is nonzero, so the inverse of matrix $\boldsymbol{B}_w$ exists, hence the SMC law for the robot is presented as (14) exists.

B. Adaptive Control Based on Neural Approximation

In this section, the RBF neural networks are trained to approximate a_1, a_2, a_3 in the matrix $\boldsymbol{A}_w$ of the control law in (14). That is the matrix containing robot's parameters such as the mass (M), the radius of the wheel (r) and the inertia moment (I_v). The RBF neural networks use gradient descent algorithms to online update the weight values.

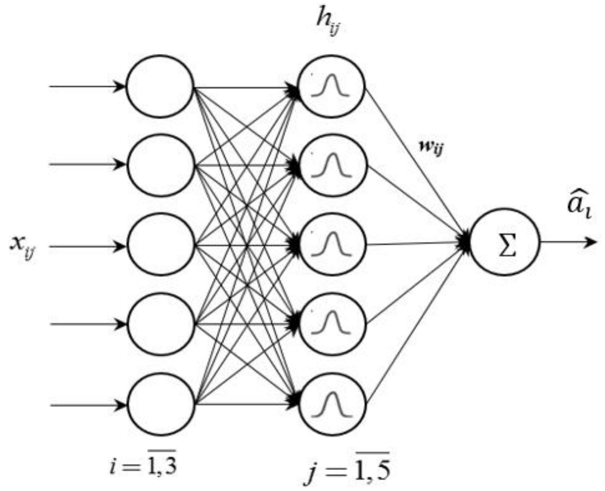

Figure 2. Block diagram of the proposed RBF neural network.

A structure [5-5-1] of the RBF neural network is used to approximate the functions $a_{i|i=1,3}$ in the matrix $\boldsymbol{A}_w$ is presented as Fig. 2. Each RBF neural network contains 5 Gaussian functions that can be described as (16) [9]:

$$h_{ij} = \exp\left(-\frac{\left\|\boldsymbol{X}_i - \boldsymbol{c}_{ij}\right\|^2}{2\boldsymbol{b}_{ij}^2}\right)\Bigg|_{i=\overline{1,3};\ j=\overline{1,5}} \quad (16)$$

where

$$\boldsymbol{X}_i = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \boldsymbol{X}_3 \end{bmatrix} = \begin{bmatrix} e(1) & \dot{e}(1) & \beta_d(1) & \dot{\beta}_d(1) & \ddot{\beta}_d(1) \\ e(2) & \dot{e}(2) & \beta_d(2) & \dot{\beta}_d(2) & \ddot{\beta}_d(2) \\ e(3) & \dot{e}(3) & \beta_d(3) & \dot{\beta}_d(3) & \ddot{\beta}_d(3) \end{bmatrix} \quad (17)$$

$$h_{ij|i=1,2,3} = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} & h_{i4} & h_{i5} \end{bmatrix} \quad (18)$$

$$w_{ij|i=1,2,3} = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} & w_{i5} \end{bmatrix} \quad (19)$$

The outputs of RBF neural network are given by (20):

$$a_i = \boldsymbol{w}_{ij}^T \boldsymbol{h}_{ij} \quad (20)$$

Hence, the approximation matrix can be calculated as:

$$\boldsymbol{A}_w = \begin{bmatrix} \boldsymbol{w}_{1j}^T\boldsymbol{h}_{1j} & -\boldsymbol{w}_{2j}^T\phi_d\boldsymbol{h}_{2j} & 0 \\ \boldsymbol{w}_{2j}^T\boldsymbol{h}_{2j} & \boldsymbol{w}_{1j}^T\boldsymbol{h}_{1j} & 0 \\ 0 & 0 & \boldsymbol{w}_{3j}^T\boldsymbol{h}_{3j} \end{bmatrix} \quad (21)$$

Now, the SMC law (14) is called the adaptive SMC based on the RBF neural networks. So that, (14) can be rewritten as (22):

$$\boldsymbol{U}_{ASMC\_RBF} = -\boldsymbol{B}_W^{-1}\begin{Bmatrix} \boldsymbol{A}_W\dot{\boldsymbol{\beta}} - \ddot{\boldsymbol{\beta}}_d + \boldsymbol{k}\dot{\boldsymbol{e}} \\ +\boldsymbol{\eta} sign(\boldsymbol{S}) + \boldsymbol{D}_f \end{Bmatrix} \quad (22)$$

When the robot's actual trajectory deviates from the reference due to the impact of control conditions such as road surface friction, changing moment of inertia, etc., then the errors $\boldsymbol{e} = \boldsymbol{\beta} - \boldsymbol{\beta}_d$ are changed. At that time, the RBF neural networks will be automatically updated, resulting in changing of $\boldsymbol{A}_w$, so that the errors can reach the minimum values. By using the RBF neural networks in control law (22), the proposed controller can adapt to the conditions of the robot.

To prove the stability, the Lyapunov function can be defined by (23):

$$\boldsymbol{V} = \frac{1}{2}\boldsymbol{S}^T\boldsymbol{S} > 0 \quad (23)$$

$$\Rightarrow \dot{\boldsymbol{V}} = \boldsymbol{S}^T\left[\boldsymbol{A}_W\dot{\boldsymbol{\beta}} - \boldsymbol{\eta}\,\mathrm{sgn}(\boldsymbol{S})\right] = -\boldsymbol{S}^T\left[\boldsymbol{A}_W\dot{\boldsymbol{\beta}} + \boldsymbol{\eta}\,\mathrm{sgn}(\boldsymbol{S})\right] \leq 0$$

where, $\boldsymbol{\eta}$ is the matrix of symmetric positive definite. So, we can conclude that $\boldsymbol{S}(t) \to 0$ at $t \to 0$, therefore, $\boldsymbol{e}(t), \dot{\boldsymbol{e}}(t) \to 0$ at $t \to 0$.

IV. Simulation Results

The block diagram of the RBF neural networks – based adaptive sliding mode control is shown in Fig. 3.

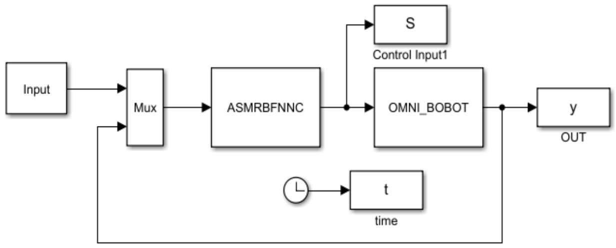

Figure 3. The block diagram of the adaptive sliding mode RBF neural network control.

The robot's parameters used for simulations include L=0.178(m), I_v=11.25 (kgm^2), M=9.4 (kg), k = 0.448, c=0.1889 (kgm^2/s), I_w= 0.02108 (kgm^2), r = 0.0245(m); and controller parameters are tuned as $\mathbf{k}=diag[25,25,58]$ and $\boldsymbol{\eta}=diag[20,20,20]$. The number of neurons in hidden layer is kept as 5 for all simulation cases.

By simulations, in order to evaluate the control quality and the system stability, several references are tested as diversity inputs. The classical SMC is also compared to the RBF neural networks–based adaptive SMC (called GD-ASMC-RBF).

Simulation results with a circle trajectory and the circle trajectory with noise are shown in Fig. 4. The quality indicators are detailed in Table I.

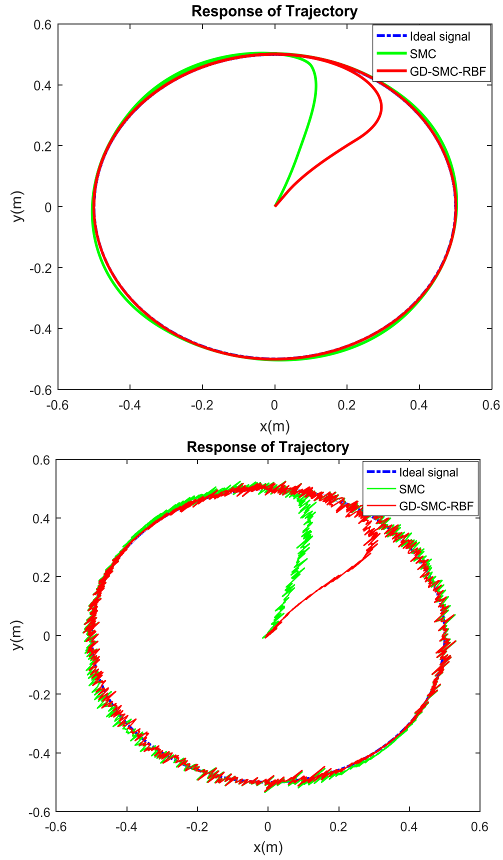

Figure 4. System responses with circle reference and noise.

TABLE I. SYSTEM RESPONSES WITH CIRCLE REFERENCE

Responses	SMC			GD-ASMC-RBF		
	POT (%)	e_{ss} (m)	t_{ss} (s)	POT (%)	e_{ss} (m)	t_{ss} (s)
x_w	0.20	0.016	0.20	0.20	0.014	0.15
y_w	0.16	0.017	0.04	0.14	0.016	0.03

In Table I, e_{ss} is the steady-state error; t_{ss} is the response time; and POT is percent of overshoot. Results show that the system responses with GD-ASMC-RBF is better than the SMC.

TABLE II. SYSTEM RESPONSES WITH ROSE REFERENCE

Responses	SMC			GD-ASMC-RBF		
	POT (%)	e_{ss} (m)	t_{ss} (s)	POT (%)	e_{ss} (m)	t_{ss} (s)
x_w	3.17	0.013	0.26	0.16	0.008	0.24
y_w	3.38	0.014	0.10	0.17	0.006	0.09

Simulation results with a rose reference and rose reference with noise are shown in Fig. 5. The quality indicators are detailed in Table II. Similarly, the system responses with GD-ASMC-RBF is better than the SMC. Especially, when adding noise, the GD-ASMC-RBF can drive the system tracking to reference with better steady-state error.

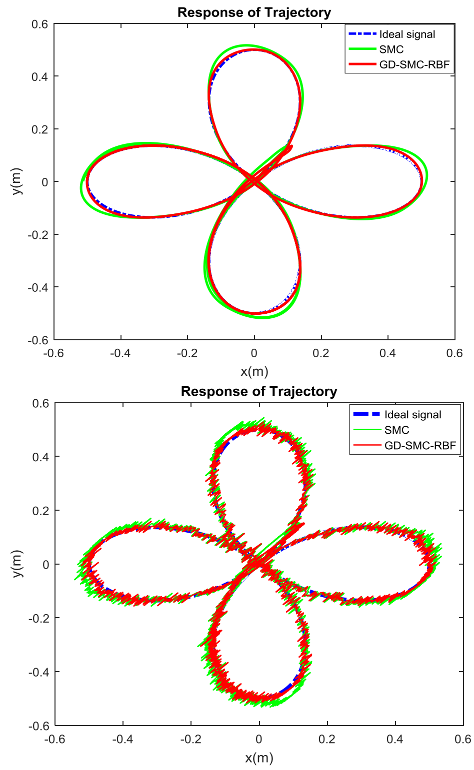

Figure 5. System responses with rose reference and noise.

The simulation results with an asteroid reference and asteroid reference with noise are shown in Fig. 6. The quality indicators are detailed in Table III. Results also show that the system responses with GD-ASMC-RBF is better than the SMC. In Fig. 6, results indicate that when the trajectory changes direction, the classical SMC can't track the reference as better as the GD-ASMC-RBF does, resulting in larger steady-state error.

TABLE III. SYSTEM RESPONSES WITH ASTEROID REFERENCE

Responses	SMC			GD-ASMC-RBF		
	POT (%)	e_{ss} (m)	t_{ss} (s)	POT (%)	e_{ss} (m)	t_{ss} (s)
x_w	2.21	0.023	0.21	3.30	0.011	0.14
y_w	2.38	0.007	0.09	0.35	0.003	0.03

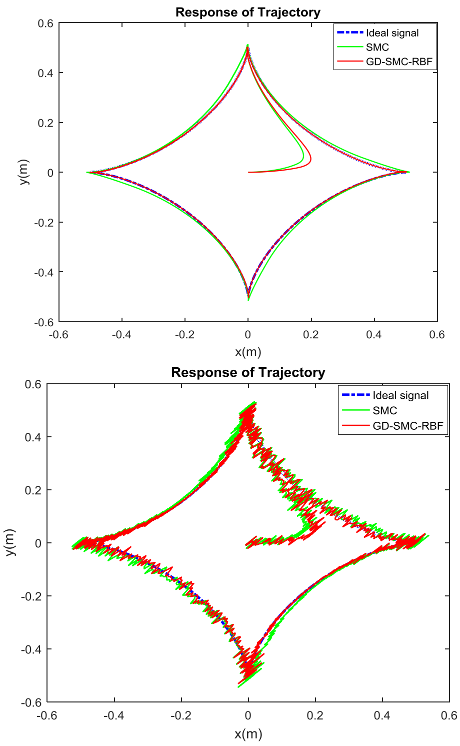

Figure 6. System responses with asteroid reference and noise.

The last simulation results with an eight-curve reference and noise are presented in Fig. 7. The quality indicators are detailed in Table IV.

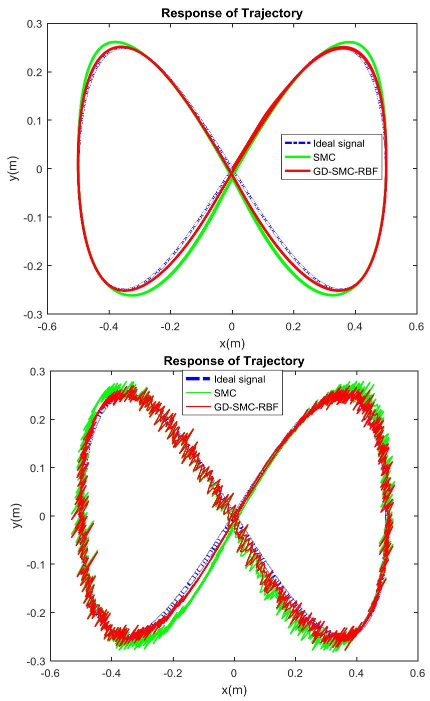

Figure 7. System responses with eight curve reference and noise.

TABLE IV. SYSTEM RESPONSES WITH EIGHT-CURVE REFERENCE

Responses	SMC			GD-ASMC-RBF		
	POT (%)	e_{ss} (m)	t_{ss} (s)	POT (%)	e_{ss} (m)	t_{ss} (s)
x_w	4.95	0.022	0.12	0.65	0.009	0.12
y_w	0.18	0.013	0.12	0.17	0.001	0.10

From above simulation results with the different inputs, results illustrate that the proposed algorithm is efficient and robust; and the responses of the system converge to desired trajectories.

V. CONCLUSION

In this paper, the control quality criteria for uncertain systems with diversity of reference signals are evaluated. The RBF neural networks – based adaptive sliding mode controller is designed and applied to control the actual trajectories of uncertain systems following the references in finite time. In addition, the omni-directional mobile robot is used to test the performance of the proposed algorithm. Moreover, the RBF neural networks are trained to approximate the robot's parameters such as the mass, the radius of the wheel and the inertia moment. The RBF neural networks are applied gradient descent training algorithms for online updating the connection weight values. The SMC law ensures that the robot's actual trajectory follows the reference, and it can adapt to the changes of control conditions by the online training mechanism for RBF neural networks. The proposed control system is simulated in MATLAB/Simulink by applying diversity inputs such as circle, rose, asteroid and eight-curve references, and white-noise as well. Simulation results indicate that by using the RBF neural networks – based adaptive SMC, the system responses are stable and robust with diversity inputs even noise affected on the system outputs, and system responses can track the references better than the classical SMC.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Dr. Thanh Tung Pham, the first author, has prepared the manuscript. Mr. Minh Thanh Le, the 2nd author has contributed on simulations. And Prof. Dr. Chi-Ngon Nguyen is chief of research group, who has supervised for this study and finalized the paper.

REFERENCES

[1] Omar Yaseen Ismael, John Hedley, "Analysis, Design, and Implementation of an Omnidirectional Mobile Robot Platform," *American Scientific Research J for Engineering, Technology, and Sciences* (*ASRJETS*), pp. 195-209, 2016.

[2] Ioan Doroftei, Victor Grosu and Veaceslav Spin, "Omnidirectional Mobile Robot – Design and Implementation," *In book: Bioinspiration and Robotics Walking and Climbing Robots*, pp. 511 – 528, 2015, "doi: 10.5772/5518".

[3] Yong Liu, J. Jim Zhu, Robert L. Williams II, Jianhua Wu, "Omni-directional mobile robot controller based on trajectory linearization," *Science Direct, Robotics and Autonomous Systems 56*, pp. 461–479, 2008, "doi: https://doi.org/10.1016/j.robot.2007.08.007".

[4] Fan Jinhui, Jia Songmin, Li Xiuzhi, "Direct Adaptive Control Based on Improved RBF Neural Network for Omni-directional Mobile Robot," *International Conference on Mechatronics, Electronic, Industrial and Control Engineering*, pp. 1108 – 1112, 2015, "doi: https://doi.org/10.2991/meic-15.2015.252".

[5] Qing Xu, Jiangming Kan, Shanan Chen and Shenqui Yan, "Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink," *International Journal of Control and Automation*, Vol.7. No.8, pp. 233 – 244, 2014, "doi: 10.14257/ijca.2014.7.8.20".

[6] Ya-Chao Yang and Chi-Cheng Cheng, "Robust sliding mode tracking control of an omnidirectional mobile robot with structural uncertainty," *Proceedings of the IASTED International Conference Modelling, Simulation, and Identification, China*, 2009.

[7] Watanabe, K., "Control of an omnidirectional mobile robot," *Second International Conference on Knowledge-Based Intelligent Electronic Systems*, pp. 51–60, 1998, "doi: 10.1109/KES.1998.725827".

[8] Jinkun Liu, *Intelligent Control Design and MATLAB Simulation*, Springer, 2018, ch 9.

[9] Jinkun Liu, *Sliding Mode Control Using MATLAB*, Academic Press, 2017, ch 6.

Thanh Tung Pham received BSc degree in EE at Mekong University in 2004, and MSc degree in Automation at Ho Chi Minh City University of Transportation in 2010. The degree of Ph.D. was award by the Ho Chi Minh City University of Transport, Vietnam, in 2019. Recently, he works at Vinh Long University of Technical Education.

Minh Thanh Le received BSc degree in EE at Mekong University in 2006, and MSc degree in Automation at Ho Chi Minh City University of Transportation in 2011. He is now a lecturer at the Faculty of Electrical and Electronic Engineering, Vinh Long University of Technical Education.

Chi-Ngon Nguyen received his BSc and MSc degree in Electronic Engineering at Can Tho University and Ho Chi Minh City University of Technology, in 1996 and 2001, respectively. The degree of Ph.D. was award by the University of Rostock, Germany, in 2007.

Since 1996, he has worked at the Can Tho University. Currently, he is an associate professor in automation and vice chairman of the board of trustees of Can Tho University. His research interests are intelligent control, medical control, pattern recognition, classifications, speech recognition and computer vision.