Training the RBF Neural Network-Based Adaptive Sliding Mode Control by BFGS Algorithm for Omni-Directional Mobile Robot

Dinh Tu. Nguyen[1], Chi Cuong Tran[1], Hoang Dang Le[1], Thanh Tung Pham[2], Chi Ngon Nguyen[1]
[1]Department of Automation Technology, Can Tho University, Vietnam
[2]Vinh Long University of Technology Education, Vietnam
E-mail: tu101074@gmail.com, {ncngon[4]}@ctu.edu.vn, {tungpt[3]}@vlute.edu.vn, { tccuong09[1]; hoangdangtdh37[2]}@gmail.com

Abstract—**This study aims to build the adaptive sliding mode control based on radial basis function neural network, thereby offering online training algorithm allows self-adjusting controller parameters according variation characteristics of nonlinear dynamic. The controller based on radial basis function network structure that is trained online using Quasi-Newton method, this method for quadratic convergernce rate is faster and more precise than the traditional Gradient Descent algorithm. Training algorithm based on radial basis function network to approximate the Hessian matrix of each training period and apply the algorithms Broyden, Fletcher, Goldfarb and Shanno to update weights in the neural network. Testing simulation through MATLAB® and experiment with Omni-directional mobile robots. The process modeling results demonstrate that the RBF trained by BFGS algorithm are fast, reliable, and accurate.**

Index Terms—**online training algorithm, adaptive sliding mode control, omni-directional mobile robot**

I. Introduction

Omni-directional mobile robot is a holonomic robot widely used in many fields and have attracted much attention of many research because of their flexible and versatile function in many application [1]. It has an advantage of moving in any abitrary course without changing the direction of wheels. The Omni-directional mobile robot model is a nonlinear MIMO system, so the problem of omni-directional mobile robots in complex trajectories is a difficult problem. In order to achieve accurare trajectory tracking and good control performance. Many research have indicated that using neural network that can be trained to learn any function. This self-learning ability of neural networks eliminate the use of complex and difficult mathematical analysis like a omni-directional mobile robot. [2]. In recent years, the analytical study of adaptive nonlinear control system using RBF universal function approximation has received much attention [3]-[5]. The RBF network is a special form of artificial neural network, which has the advantages of simpler structure, faster algorithms, and better approximation of a nonlinear relation [3], [6], [7]. Radical basis function theory allows the optimization of the neural network's weight over the whole space of the state variable but ensures the online learning process. To supplement and overcome the limitations of controlling the Omni-directional mobile robot in given trajectories, the paper proposes a solution using radial basis function (RBF) neural network and solves the problem.

There are three common learning methods: supervised learning, unsupervised learning and reinforcement learning. Supervised learning is the most commonly used method, the simplest being the gradient descent algorithm. But the algorithm is simple and low convergence To improve convergence faster, the BFGS algorithm is introduced. BFGS algorithm is a Quasi- Newton algorithm which helps in faster convergence and it also takes care of the inverse of Hessian matrix. This method approximates the Hessian matrix using recent function and gradient evalutions which Broyden, Fletcher, Goldfarb and Shanno (BFGS) algorithm to update the weights for networks with simple structure like Rwill give you very fast convergence. Therefore, a complex nonlinear model with parametric uncertainties such as the Omni- directional mobile robot, it is highly recommended to provide an online training algorithm using a simple structured neural network such as RBF.

II. Mathematical Modelling of Omni-Directional Mobile Robot

By Newton's law, the dynamics of omni-directional mobile robot under this assumtion can be found in [8]-[11]. There are two coordinate system referred to this modeling. The absolute coordinate system $O_w - X_w Y_w$ and the moving coordinate system $O_m - X_m Y_m$ are shown in Fig. 1.

Define $S_w = [X_w \quad Y_w]^T$ to be the position vector of the center of gravity for the mobile robot, we have

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

Manuscript received January 14, 2018; revised April 23, 2018.

doi: 10.18178/ijmerr.7.4.367-373

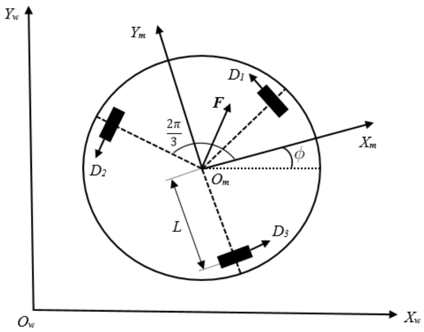

Figure 1. Absolute coordinate and force analysis.

where M is a symmetric positive-definite matrix as $M = diag\ \ m, m$ with the mass m.

When indtroducing the coordinate transformation matrix from absolute coordinate system to moving coordinate system such as (4) and it follows (3), (4).

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

$$ \dot{S}_w = {}^wR_m \dot{s}_m \tag{3} $$

$$ F_w = {}^wR_m f_m \tag{4} $$

where

$s_m = \ X_m \quad Y_m\ ^T$, $f_m = \left[f_x \quad f_y \right]^T$, $F_m = \left[F_x \quad F_y \right]^T$ are the position vector of center of gravity and the force vector applied to the center of gravity in the moving coordinate system and the force vector applied to center of gravity in the absolute coordinate system respectively. Let ϕ denote the angle between X_w and X_m.

Therefore, the equation (5) to be a transformating equation to the moving coordinate system.

$$ M \ \ {}^wR_m^T\, {}^w\dot{R}_m \dot{s}_m + \ddot{s}_m \ \ = f_m \tag{5} $$

The dynamic properties can be described as [12]

$$ M(\ddot{X}_m - \dot{Y}_m \dot{\phi}) = f_x \tag{6} $$

$$ M(\ddot{Y}_m + \dot{X}_m \dot{\phi}) = f_y \tag{7} $$

$$ I_v \ddot{\phi} = M_I \tag{8} $$

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

$$ f_x = -\frac{1}{2} D_1 - \frac{1}{2} D_2 + D_3 \tag{9} $$

$$ f_y = \frac{\sqrt{3}}{2} D_1 - \frac{\sqrt{3}}{2} D_2 \tag{10} $$

$$ M_I = (D_1 + D_2 + D_3)L \tag{11} $$

Besides, the driving system property [13] for each assembly is assumed to be given by

$$ I_w \dot{w}_i + c w_i = k u_i - r D_i \ (i = 1, 2, 3) \tag{12} $$

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

On the other hand, the geometrycal relationships among variable $\dot{\phi}$, $\dot{X}_m$ $\dot{Y}_m$ and ω_i ,i.e., the inverse kinematics are given by

$$ r\omega_1 = -\frac{1}{2}\dot{X}_m + \frac{\sqrt{3}}{2}\dot{Y}_m + L\dot{\phi} \tag{13} $$

$$ r\omega_2 = -\frac{1}{2}\dot{X}_m - \frac{\sqrt{3}}{2}\dot{Y}_m + L\dot{\phi} \tag{14} $$

$$ r\omega_3 = \dot{X}_m + L\dot{\phi} \tag{15} $$

Therefore, we use equation (6)~(15) gives

$$ \ddot{X}_m = a_1 \dot{X}_m + a_2' \dot{Y}_m \dot{\phi} - b_1(u_1 + u_2 - 2u_3) \tag{16} $$

$$ \ddot{Y}_m = a_1 \dot{Y}_m - a_m' \dot{X}_m \dot{\phi} + \sqrt{3} b_1 (u_1 - u_2) \tag{17} $$

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

where,

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

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

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

$$ b_1 = \frac{kr}{(3I_w + 2Mr^2)} $$

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

In this paper, the tracking control problem for the omni-directional mobile robot, which allows the position of robot platform can be kept at the reference position and makes the tracking error converging to zero in the of uncertainty is resolved. An adaptive sliding mode control using RBF is presented to account for the parameter uncertainty.

The control object is to design the reference signal to the tracking control. Therefore, the equation is given by

$$\begin{bmatrix}\ddot{X}_w\\ \ddot{Y}_w\\ \ddot{\phi}\end{bmatrix}=\begin{bmatrix}a_1 & -a_2\phi_d & 0\\ a_2\phi_d & a_1 & 0\\ 0 & 0 & a_3\end{bmatrix}\begin{bmatrix}X_w\\ Y_w\\ \phi\end{bmatrix}+\begin{bmatrix}b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\phi\\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\phi\\ b_2 & b_2 & b_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix}+\begin{bmatrix}D_{fx}\\ D_{fy}\\ D_{f\phi}\end{bmatrix}=A_W\beta+B_WU+D_f$$

The tracking error is defined by

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

where $\beta=[X_w \quad Y_w \quad \phi]^T$ and $\beta_d=[X_d \quad Y_d \quad \phi_d]^T$ are the output trajectory position and the desired trajectory position, respectively.

The sliding mode surface function is defined as

$$S=\begin{bmatrix}(\dot{X}_w-\dot{X}_d)+k_x(X_w-X_d)\\ (\dot{Y}_w-\dot{Y}_d)+k_y(Y_w-Y_d)\\ (\dot{\phi}-\dot{\phi}_d)+k_\phi(\phi-\phi_d)\end{bmatrix}=\begin{bmatrix}\dot{e}_x+k_xe_x\\ \dot{e}_y+k_ye_y\\ \dot{e}_\phi+k_\phi e_\phi\end{bmatrix}=\begin{bmatrix}S_x\\ S_y\\ S_\phi\end{bmatrix},$$

where $k=diag\left[k_x \quad k_y \quad k_\phi\right]$ is the positive constants and must satisfy Hurwiz condition. The sliding surface and friction function are presented in $S=\left[S_x \quad S_y \quad S_\phi\right]^T$ and $d(t)$, respectively.

III. Controller Design for Trajectory Tracking

A. Network Architecture

The structure of a typical three-layer RBF neural network is shown as Fig. 2. In RBF neural network, $x=[x_i]^T$ is input vector. Assuming there are m^{th} neural nets, and radial basis function vector in hidden layer of RBF is $h=[h_i]^T$, h_i is Gaussian function value for neural net j in hidden layer, and

$$h_j=\exp\left(-\frac{\left\|x-c_j\right\|^2}{2b_j^2}\right), \tag{19}$$

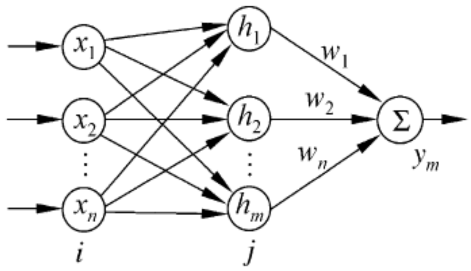

Figure 2. RBF neural network structure [14].

The weight value of RBF is

$$w=[w_1,...,w_m]^T \tag{20}$$

The output of RBF neural network is

$$y(k)=w_0+w_1h_1+w_2h_2+...+w_mh_m=w_0+\sum_{j=1}^{m}w_jh_j$$

B. Trajectory Controller Design

The adaptive sliding mode control [15]-[17] is presented to solve the difficulty caused by parameter uncertainties and disturbances. In this part, this study uses RBF neural network to approximate the A_W component. Then, let the control law be defined as

$$U=-B_W^{-1}\left(A_W\dot{\beta}+\ddot{\beta}_d+k\dot{e}+\eta\,\mathrm{sgn}(S)\right), \tag{21}$$

where, $\eta=\begin{bmatrix}\eta_x & 0 & 0\\ 0 & \eta_y & 0\\ 0 & 0 & \eta_\phi\end{bmatrix}$, $\Gamma=A_W$, with the parameter estimate error being defined as

$$\Gamma=\Gamma-\hat{\Gamma}, \tag{22}$$

where

$$\hat{\Gamma}=\begin{bmatrix}f_1 & -f_2\phi_d & 0\\ f_2\phi_d & f_1 & 0\\ 0 & 0 & f_3\end{bmatrix}=\begin{bmatrix}W^Th_1(x) & -V^T\phi_dh_2(x) & 0\\ V^Th_2(x) & W^Th_1(x) & 0\\ 0 & 0 & Z^Th_3(x)\end{bmatrix}$$

where, W^T,V^T,Z^T are the weight values of RBF , ε is the tracking error of RBF neural network, and $\varepsilon\leq\varepsilon_N$. Then, the time derivative of S become.

$$\dot{S}=\ddot{e}+k\dot{e}=\ddot{\beta}_d-\ddot{\beta}+k\dot{e}$$
$$=\ddot{\beta}_d-(A_W+\Delta A_W)\begin{bmatrix}\dot{X}_w\\ \dot{Y}_w\\ \dot{\phi}\end{bmatrix}-B_W(\phi)\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix}-d(t) \tag{23}$$
$$=-\eta\,\mathrm{sgn}(S)-d(t).$$

Therefore, if $\eta \geq D$, we have

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

Then, from (21) and (23), $\dot{S}$ can be described as

$$\begin{aligned}\dot{S} &= \ddot{\beta}_d - A_W\dot{\beta} - B_W U - d(t) + k\dot{e} \\ &= \ddot{\beta}_d - A_W\dot{\beta} - (-A_W\dot{\beta} + \ddot{\beta}_d + k\dot{e} + \eta\,\mathrm{sgn}(S)) - d(t) + k\dot{e} \\ &= -A_W\dot{\beta} + A_W\dot{\beta} - \eta\,\mathrm{sgn}(S) - d(t) \\ &= -A_W\dot{\beta} - d(t) - \eta\,\mathrm{sgn}(S)\end{aligned}$$

where $\Gamma = \Gamma - \hat{\Gamma}$, $\Gamma = A_W$, $\hat{\Gamma} = A_W$. The Lyapunov candidate function can be defined as (24).

$$V = \frac{1}{2}S^T S > 0 \qquad (24)$$

Then, the time derivative of V become.

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

where η is the positive definite symmetric matrix. Moreover, the tracking error $e(t)$ will converge to zero according to $S(t) \to 0$ as $t \to \infty$ asymptotically and therefore, $e(t), \dot{e}(t) \to 0$ as $t \to \infty$.

C. Gradient Descent with Momentum Training Algorithm

The performance index function of RBF is

$$E = \frac{1}{2}(y(k) - y_m(k))^2 \qquad (25)$$

According to gradient descent method, the parameters can be updated as follows.

$$\Delta w_j(t) = -\eta\frac{\partial E}{\partial w_j} = \eta(y(t) - y_m(t))h_j \qquad (26)$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha(w_j(t-1) - w_j(t-2))$$

where $\eta \in (0,1)$ is the learning rate and $\alpha \in (0,1)$ is the momentum factor.

D. BFGS Training Algorithm

For an unconstrained optimal problem [18], [19], the conventional Newton algorithm or Quasi-Newton (BFGS) is based on a quadratic model from a truncated Taylor series expansion of the objective function around w_k

$$E(w_k + \delta) = E_k(\delta_k) = E_k + g_k^T\delta + \frac{1}{2}\delta^T H_k\delta. \qquad (27)$$

where $E_k(\delta)$ is the resulting quadratic approximation at iteration k, $\delta = w - w_k$, $g_k = \nabla E_k = \frac{\partial E}{\partial w_k}$ is Gradient of the objective function [20], and $G_k = \nabla^2 E_k$ is the Hessian matrix [20]. In fact, approximation of Hessian in BFGS method is important. Let $H_0, H_1, H_2...$ be succesive approximations of inverse G_k^{-1} of the Hessian.

Quasi- Newton algorithm is given by follow form:

$$\begin{aligned}d_k &= -H_k g_k \\ \alpha_k &= \arg\min E(w_k + \alpha d_k), \\ w_{k+1} &= w_k + \alpha_k d_k.\end{aligned}$$

Quasi-Newton (BFGS) algorithm [19] including 4 steps with k iteration:

- *Step 1 :* Set the search direction.

$$d_k = -H_k^{-1}g_k. \qquad (28)$$

- *Step 2 :* Search along the direction d_k to find the step length α_k such that

$$f(w_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(w_k + \alpha_k d_k). \qquad (29)$$

$$\begin{aligned}&E(w_k + \alpha_k d_k) \leq E(w_k) + \alpha_k g_k^T d_k \\ &\left|g(w_k + \alpha_k d_k)^T\right| \leq -g_k^T d_k\end{aligned} \qquad (30)$$

The new weighting vector is updated as

$$w_{k+1} = w_k + \alpha_k d_k. \qquad (31)$$

- *Step 3 :* Update H_k to H_{k+1}.

$$H_{k+1} = H_k + \frac{\Delta g_k \Delta g_k^T}{\Delta g_k^T \delta_k} - \frac{H_k\delta_k\delta_k^T H_k}{\delta_k^T H_k \delta_k} \qquad (32)$$

where

$$\delta_k = w_{k+1} - w_k, \qquad (33)$$

$$\Delta g_k = g_{k+1} - g_k. \qquad (34)$$

- *Step 4 :* Check convergence by some specified criterion. Conventionally, this criterion is set as the gradient of the objective function to be smaller than a given value to stop the iteration.

$$\sqrt{g_k^T g_k} \leq \varepsilon. \qquad (35)$$

The iteration starts from letting $k = 0$, while the initial values of components in the weighting vector are set randomly, the initial matrix H_0 is a symetric and positive-definite matrix that is normally set to $H_0 = I$. The iteration process sequentially follows: to calculate first the search direction by (28) and then perform the inexact search based on (30) and H_k to H_{k+1} by (32), (33) and (34), and after that, the gradient g_k is compared with the convergence criterion, as shown in (35). When the convergence criterion is satisfied, the training iteration ends, otherwise, set $k = k+1$ and repeat the whole iteration calculation.

IV. Simulation Results

In this part, the proposed adaptive sliding mode control using RBF neural network approach will be applied to the tracking problem of the omni-directional mobile robot. It was assumed that the initial position was $0 \quad 0 \quad 0$ for all test. All physical parameters of the model and the default parameter are as follows:

$$I_v = 11.25 kgm^2,\ m = 9.4 kg$$
$$L = 0.178 m,\ K = 0.448,\ c = 0.1889 kgm^2/s$$
$$I_\omega = 0.02108 kgm^2,\ r = 0.0245 m$$

TABLE I. Controller Parameter

	Parameter	Value
Sliding mode	$\eta = \begin{bmatrix} \eta_x & 0 & 0 \\ 0 & \eta_y & 0 \\ 0 & 0 & \eta_\phi \end{bmatrix}$	$\begin{bmatrix} 20 & 0 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 20 \end{bmatrix}$
	$k = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\phi \end{bmatrix}$	$\begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 58 \end{bmatrix}$

A. Without Friction

In the first case, the friction function and the model uncertainties are not considered. The tracking results including the tracking error, and the robot trajectory are shown in Fig. 3. The controller parameters are given as in Table 1 and the reference trajectory in the test in a circle defined as follows:

$$\begin{cases} x = 0.3\cos\ 2\pi t \\ y = 0.3\sin\ 2\pi t \\ \phi = \dfrac{2}{3}\pi \end{cases} \tag{36}$$

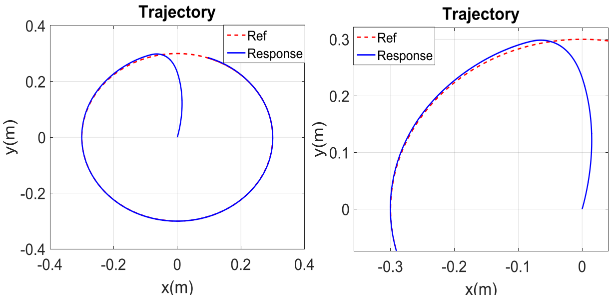

Figure 3. Tracking performances for circular reference path without friction.

The omni-directional mobile robot was stationary at the initial position. In Fig. 3 showns the trajectory tracking responses, the system tracking trajectory finally converged to the reference circle.

B. With Friction

In this case, the friction are considered, and the tracking results are depicted in Fig. 4.

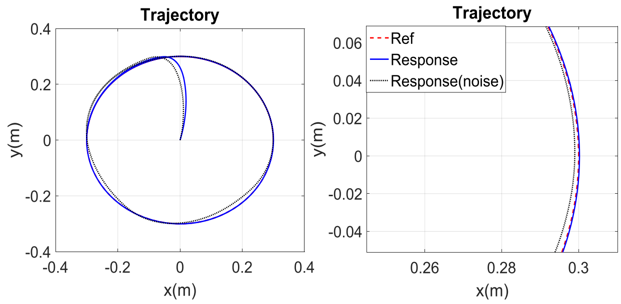

Figure 4. Tracking performances for a circular reference path with friction.

Fig 4 apparently shows that the system tracking trajectory was able to follow the reference circle with a noise range of 0.01.

C. Compare with Gradient Descent (Momentum)

In this case, the comparison between the BFGS algorithm and Gradient descent with momentum are considered, and the results are depicted in Fig. 5, Fig. 6, Fig. 7 and Fig.8.

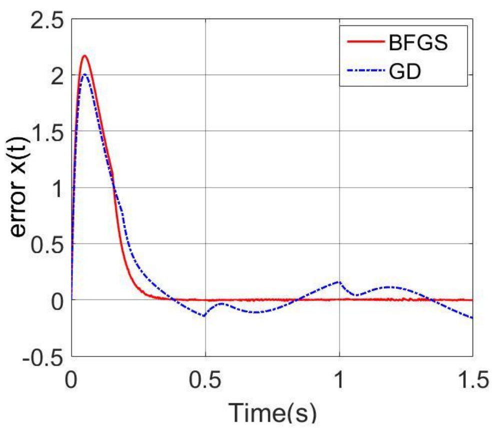

Figure 5. Tracking errors for x.

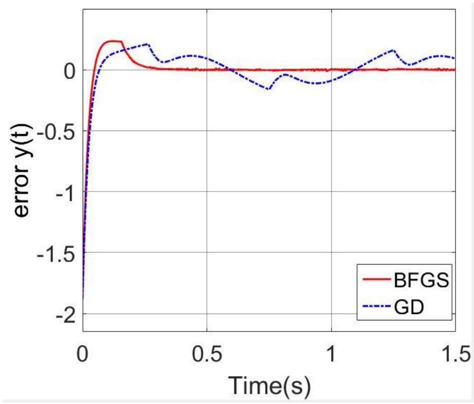

Figure 6. Tracking error for y .

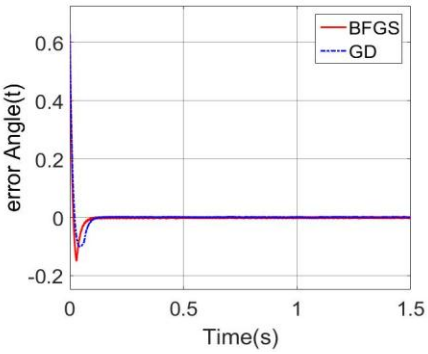

Figure 7. Trackingn error for ϕ .

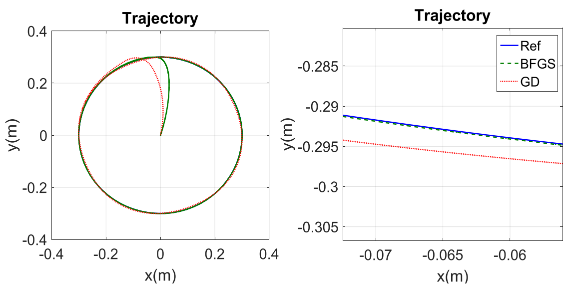

Figure 8. Tracking performance for comparison between BFGS and Gradient descent with momentum.

The tracking error results are described in Fig. 5, Fig. 6 and Fig. 7. The error states did not converge to the zero by the adaptive sliding mode using Gradient descent algorithm. Nevertheless, the error states are reduced to zero by the adaptive sliding mode control using BFGS algorithm.

Fig. 8 apparently shows that the system tracking trajectory did not respectively converge to the reference circle by adaptive sliding mode control using Gradient descent algorithm. Neverthless, by the adaptive sliding mode control using Quasi-Newton with BFGS, the system tracking trajectory was able to follow the reference circle.

V. Conclusion

In this paper, the adaptive sliding mode controller using RBF neural network to approximate parametric uncertainties which is developed to achieve perfect tracking for omni-directional mobile robot. The nonlinear friction including disturbance and traction forces is considered to reflect actual dynamic behavior of the robot in the real world. According to the simulation results, the proposed controller can effectively overcome possible uncertainties and external disturbances. The results also show that adaptive sliding mode controller takes more time to stabilize when error minimization is done using Gradient descent with momentum and it show poor tracking of reference signal. But after applying the BFGS learning the tracking behavior of system improved considerably and it shows less tracking error. BFGS method is faster method as it assumes Hessian matrix approximation as Identity matrix.

References

[1] K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang, and T. Fukuda, "Feedback control of an omnidirectional autonomous platform for mobile service robots," *Journal of Intelligent and Robotic Systems*, vol. 22, pp. 315-330, 1998.

[2] T. Hayakawa, W. M. Haddad, N. Hovakimyan, and V. Chellaboina, "Neural network adaptive control for nonlinear nonnegative dynamical systems," *IEEE Transactions on Neural Networks*, vol. 16, pp. 399-413, 2005.

[3] A. Al-Amoudi and L. Zhang, "Application of radial basis function networks for solar-array modelling and maximum power-point prediction," in *Proc. Generation, Transmission and Distribution*, 2000, pp. 310-316.

[4] A. R. Barron, "Approximation and estimation bounds for artificial neural networks," *Machine Learning*, vol. 14, pp. 115-133, 1994.

[5] T. Chen and H. Chen, "Approximation capability to functions of several variables, nonlinear functionals, and operators by radial basis function neural networks," *IEEE Transactions on Neural Networks*, vol. 6, pp. 904-910, 1995.

[6] V. Elanayar and Y. C. Shin, "Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems," *Neural Networks, IEEE Transactions on*, vol. 5, pp. 594-603, 1994.

[7] K. Warwick and R. Craddock, "An introduction to radial basis functions for system identification: A comparison with other neural network methods," 1997.

[8] J. E. M. Salih, M. Rizon, S. Yaacob, A. H. Adom, and M. R. Mamat, "Designing omni-directional mobile robot with mecanum wheel," *American Journal of Applied Sciences*, vol. 3, pp. 1831-1835, 2006.

[9] M. Wada and H. H. Asada, "Design and control of a variable footprint mechanism for holonomic omnidirectional vehicles and its application to wheelchairs," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 978-989, 1999.

[10] M. Wada, A. Takagi, and S. Mori, "Caster drive mechanisms for holonomic and omnidirectional mobile platforms with no over constraint," in *Proc. 2000 Robotics and Automation, ICRA'00, IEEE International Conference on*, 2000, pp. 1531-1538.

[11] J. Wu, R. L. Williams, and J. Lew, "Velocity and acceleration cones for kinematic and dynamic constraints on omni-directional mobile robots," *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, pp. 788-799, 2006.

[12] M. Iwatsuki, K. Nakano, and T. Ohuchi, "Target point tracking control of robot vehicle by fuzzy reasoning," *Transactions of the Society of Instrument and Control Engineers*, vol. 27, pp. 70-76, 1991.

[13] M. Saito and T. Tsumura, "Collision free motion control of multiple mobile robots on path network," in *Proc. Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, 1991, pp. 1068-1073.

[14] J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation*: Springer Science & Business Media, 2013.

[15] J. Wang, J. Wang, and T. Zhang, "RBF neural network based adaptive sliding mode control for hypersonic flight vehicles," in *Proc. 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, , 2016, pp. 58-63.
[16] Z. Jie, H. Long, and R. Sijing, "RBF neural network adaptive sliding mode control based on genetic algorithm optimization," in *Proc. 2016 Chinese Control and Decision Conference (CCDC)*, 2016, pp. 6772-6775.
[17] H. Tongyue and F. Juntao, "Adaptive sliding mode control of flexible beam using RBF neural controller," in *Proc. 2015 34th Chinese Control Conference (CCC)*, 2015, pp. 3405-3410.
[18] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: Theory and Algorithms*, John Wiley & Sons, 2013.
[19] D. G. Luenberger, "Introduction to linear and nonlinear programming," vol. 28, Addison-Wesley Reading, MA, 1973.
[20] S. J. Wright and J. Nocedal, "Numerical optimization," *Springer Science*, vol. 35, p. 7, 1999.

Dinh Tu Nguyen received his B.S.degree in electronic and communication engineering from Can Tho University, Can Tho city, Vietnam, in 2015. Since 2015, he has been been working toward a M.S degree in the Department of Automation Technology, Can Tho University. His current research interests include linear and nonlinear control theories, computer vision, image processing and intelligent control to mechatronic system.

Chi Cuong Tran received his B.S. degree in Control Engineering and Automation from Can Tho University, Can Tho city, Vietnam, in 2015. Since 2015, he has been working toward a M.S degree in the Department of Automation Technology, Can Tho University. His current research interests include intelligent control and computer vision.

Hoang Dang Le received his B.S. degree in control engineering and automation from Can Tho University, Can Tho city, Vietnam, in 2015. Since 2015, he has been working toward a M.S degree in the Department of Automation Technology, Can Tho University. His current research interests include intelligent control and system identification.

Thanh Tung Pham received his B.S. and M.S. degree in electrical and electronic engineering from Mekong University and Ho Chi Minh City University of Transport, in 2004 and 2010, respectively. Currently, he is a senior lecture in the Faculty of Electrical and Electronic Engineering Technology, Vinh Long University of Technology Education.

Chi Ngon Nguyen received his B.S. and M.S. degree in electrical engineering from 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 Can Tho University. Currently, he is a senior lecture in the Department of Automation Technology. He is working as position of director of Electrical and Electronic Center, and dean of the College of Engineering Technology at the Can Tho University. His research interests are intelligent control, medical control, pattern recognition, classifications, speech recognition and computer vision.