# Dynamic Control of Mobile Robot Using RBF Global Fast Sliding mode

**Ali Mallem[1], Noureddine Slimane[2], Walid Benaziza[3]**
[1,3]Department, Batna 2 University, street Chahid Boukhlouf M$^{ed}$El Hadi Batna, Algeria
[2]Advanced Electronics Laboratory, Faculty of Engineering, University of Batna 2, Algeria

**Article Info**

***Article history:***

Received Jan 28, 2018
Revised May 20, 2018
Accepted Jun 5, 2018

***Keyword:***

Dynamic model
Global fast sliding mode
Lyapunov stability
RBF neural network

**ABSTRACT**

This paper mainly in this paper a dynamic control of mobile robot using RBF global fast sliding mode (RBF-GFSM) strategy is presented. Firstly, a GFSM controller is used in order to make the linear and angular velocities converge to references ones in finite time. However, a problem of instability of velocities is appeared by introducing disturbances in the system. Secondly, a combined controller using RBF-GFSM approach is applied in aim to stabilize the velocities errors and estimates the nonlinear function of the robot model. The system stability is done using the lyapunov theory. The proposed controllers are dynamically simulated using Matlab/Simulink and the simulations results show the efficiency and robustness of the proposed control strategy.

***Corresponding Author:***

Ali Mallem,
Department, Batna 2 University,
street Chahid Boukhlouf M$^{ed}$El Hadi, Batna, Algeria
Email: ali_mallem@hotmail.fr

## 1. INTRODUCTION

Mobile robots study has attracted important advantage in the robotics and control research community, due to the nonholonomic properties caused by not integrable differential constraints. The mobile robot problem is the motion under nonholonomic constraints using the kinematic model and specially the problem of integration of the nonholonomic kinematic controller with the dynamics of the mobile robot [1]. Mobile robot navigation can be classified into three basic problems [2]; reference trajectory tracking, path following, and situation stabilization. Some nonlinear feedback controllers have been proposed for solving these problems [2]-[4]. The main idea behind these algorithms is to find suitable velocity control inputs which stabilize the closed-loop system.

In recent years, different control techniques have been introduced to control mobile robot. Due to the intrinsic nonlinearity in the mobile robot dynamics and the nonholonomic constraints, nonlinear architectures as adaptive and intelligent methods [5]-[7], backstepping [8], [9] feedback linearization [10] and sliding mode control [11] have been studied. The neural network controller can deal with no modeled bounded disturbances and/or unstructured no modeled dynamics of the mobile robot. Therefore, a control structure that makes possible the integration of a kinematic controller and a neural network (NN) computed-torque controller for nonholonomic mobile robots is presented in [12]. A neuro-fuzzy network (NFN) dynamic controller for mobile robots is presented in [13], with a combined kinematic/dynamic control law is developed using backstepping and stability is guaranteed by Lyapunov theory.

An adaptive neural conventional sliding mode controller for nonholonomic wheeled mobile robots with model uncertainties and external disturbances is presented in [14]. In this work the kinematic model is presented by the polar coordinates and dynamic model with uncertainties is considered. Self recurrent wavelet neural networks (SRWNNs) are used for approximating the model uncertainties and deal disturbances. In new researches of tracking control, a finite time tracking control was deployed in last year's [15]. A global finite-time tracking controller was given for the nonholonomic systems in [16]. The finite-time tracking control is presented in [17], [18].

For the differential equation of time analysis, the literature proposed finite time stability. So, to study a class of the system stability for a limited time becomes reasonable [19]-[22]. The paper is organized as follows. Section II presents the mobile robot dynamic modeling. Section III resumes RBF neural network. The RBF-GFSM controller is presented in section IV. The simulation and analysis of the improved algorithm are presented in Section V. Finally, conclusions are drawn in Section VI.

## 2. MOBILE ROBOT MODELING

The mobile robot modeling consists in two models: kinematic and dynamic models. The robot's kinematics is defined by (1):

$$\dot{p} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} q \qquad (1)$$

where q represents the control vector (v,ω)T.

Generally dynamic modeling is the system motion study in which forces are modeled and it can include energies and the speeds associated with the motions. The general dynamic model of mobile robot can be described by the following equation (2):

$$M(q)\ddot{q} + V(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) + \tau_d = B(q)\tau - A^T(q)\lambda \qquad (2)$$

where M(q) is the symmetric positive definite inertia matrix, $\mathrm{V}(\mathrm{q},\dot{\mathrm{q}})$ is the centripetal and coriolis matrix, $\mathrm{F}(\dot{\mathrm{q}})$ is the surface friction matrix, G(q) is the gravitational vector, τd denoted bounded unknown disturbances including unstructured not modeled dynamics, B(q) is the input transformation matrix, τ is the input vector, AT(q) is the matrix associated with the constraints, λ is the constraint forces vector. The above system can be transformed into a more suitable representation for control and simulation purposes. The two following matrices are defined to do this transformation as shown in (3-4):

$$\vartheta = \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad (3)$$

$$S(q) = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \qquad (4)$$

The matrix S (q) has the following relation with matrix A(q) (5):

$$S^T(q)A^T(q) = 0 \qquad (5)$$

The equation (2) can be rewritten as shown in (6-10):

$$Mh(q)\dot{\vartheta} + Vh(q,\dot{q})\vartheta + \tau_h = Bh(q)\tau \qquad (6)$$

with:

$$Mh(q) = S^T(q)M(q)S(q) \qquad (7)$$

$$\tau_h = S^T(q)\tau_d \tag{8}$$

$$Bh(q) = S^T(q)B(q) \tag{9}$$

$$Vh = S^T(q)M(q)\dot{S}(q) + S^T(q)V^T(q,\dot{q})S \tag{10}$$

Equation (6) is the equation which is used for the control and simulation analysis of the robot. The dynamic modeling of the robot is presented in [23].

## 3. RBF NEURAL NETWORK

RBF networks are adaptively used to approximate the uncertain nonlinear function. The algorithm of a radial basis function (RBF) networks is defined in [24] as shown in (11):

$$h_j = g(x - c_{ij}{}^2)/b_j^2) \tag{11}$$

$$f = W^t h(x) + \varepsilon$$

where $x$ is the input state of the network, $i$ is the input number of the network, $j$ is the number of hidden layer nodes. In the network, $h= [h_1\ h_2\ \ldots\ldots\ h_n]^T$ is the output of Gaussian function. $W$ is the neural network weights, and the propagation error is $\varepsilon \le \varepsilon_N$. RBF network approximation f is used. In Figure 1 is represents an RBF network. The output of RBF network is (12):

$$\hat{f}(x) = \hat{W}^T h(x) \tag{12}$$

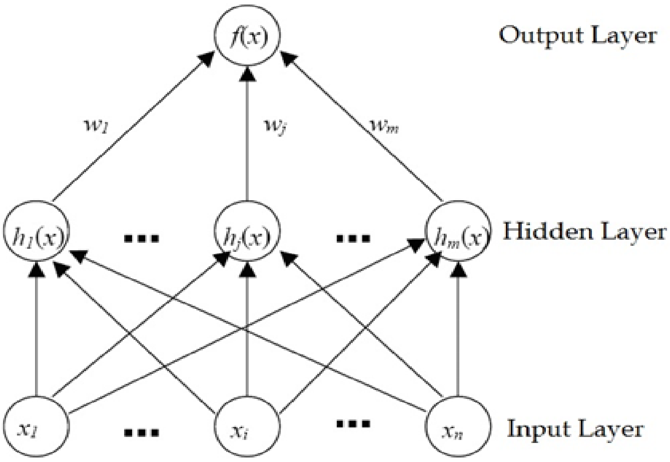

Figure 1. RBF neural network.

The Gaussian function can define as shown in (13):

$$h(x) = \exp\ (-\frac{r^2}{2\sigma^2}) \tag{13}$$

## 4. RBF-GFSM-CONTROLLER

In this work two cases of control are proposed, the first without disturbances, the second in the presence of disturbances. In Figure 2 below resumes the control strategy proposed in this work. The figure below resumes the control strategy proposed in this work.

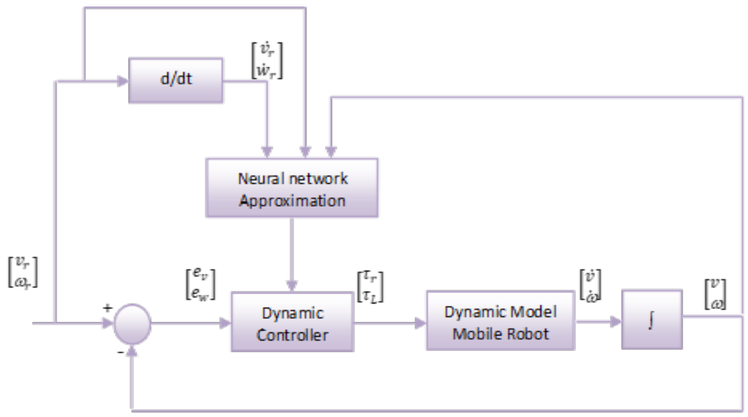
$\begin{bmatrix} \dot{v}_r \\ \dot{w}_r \end{bmatrix}$
d/dt
Neural network Approximation
$\begin{bmatrix} v_r \\ \omega_r \end{bmatrix}$
+
-
$\begin{bmatrix} e_v \\ e_w \end{bmatrix}$
Dynamic Controller
$\begin{bmatrix} \tau_r \\ \tau_L \end{bmatrix}$
Dynamic Model Mobile Robot
$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix}$
∫
$\begin{bmatrix} v \\ \omega \end{bmatrix}$

Figure 2. Control strategy

### 4.1. Control without disturbances

In the case of disturbances absence, the equation (6) becomes (14):

$$Mh(q)\dot{\vartheta} + Vh(q,\dot{q})\vartheta = Bh(q)\tau \tag{14}$$

The velocities errors are definite as shown in (15):

$$e_{\vartheta} = \vartheta_r - \vartheta = \begin{bmatrix} e_v \\ e_{\omega} \end{bmatrix} = \begin{bmatrix} v_r - v \\ \omega_r - \omega \end{bmatrix} \tag{15}$$

The derivative of (15) is obtained as shown in (16):

$$\dot{e}_{\vartheta} = \dot{\vartheta}_r - \dot{\vartheta} \tag{16}$$

According to the equation (16), the equation (14) can be rewritten as shown in (17):

$$\begin{aligned} &Mh(q)(\dot{\vartheta}_r - \dot{e}_{\vartheta}) + Vh(q,\dot{q})(\vartheta_r - e_{\vartheta}) = Bh(q)\tau \\ &Mh(q)\dot{\vartheta}_r - Mh(q)\dot{e}_{\vartheta} + Vh(q,\dot{q})\vartheta_r - Vh(q,\dot{q})e_{\vartheta} = Bh(q)\tau \end{aligned} \tag{17}$$

Putting (18):

$$f\left(x\right) = Mh\left(q\right)\dot{\vartheta}_r + Vh\left(q,\dot{q}\right)\vartheta_r \tag{18}$$

with $x = [\vartheta_r^T \ \dot{\vartheta}_r^T \ \vartheta^T \ ]^T$

Replacing (18) in (17) and as shown in (19):

$$-Mh\left(q\right)\dot{e}_{\vartheta} - Vh\left(q,\dot{q}\right)e_{\vartheta} + f\left(x\right) = Bh(q)\tau \tag{19}$$

The sliding mode controller proposed in this work is based on global fast sliding mode control, this control can make the system states converge to zero in a finite time. A kind of fast terminal sliding surface is proposed as shown in (20)**:**

$$s = \dot{x} + \alpha x + \beta x^{q/p} = 0 \tag{20}$$

Where $x \in R$ is the state and $\alpha > 0$.

The reaching time of the sliding surface to zero is defined as shown in (21):

$$t_s = \frac{p}{\alpha(p-q)} \ln \frac{\alpha x(0)^{\frac{p-q}{p}} + \beta}{\beta} \tag{21}$$

The global fast sliding surface is selected as shown in (22):

$$s = \dot{s}_0 + \alpha s_0 + \beta s_0^{q/p} \tag{22}$$

Where β>0 and *q, p* (*q*<*p*) are positives odd numbers.
Then, to obtain the control law, a sliding surface is choosing (23):

$$s = e_\vartheta \tag{23}$$

with: $s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$

According to the equations (23) and (22), one can have (24):

$$\dot{e}_\vartheta = -(\alpha+1)e_\vartheta - \beta e_\vartheta^{q/p} \tag{24}$$

Replacing (24) in (19) and as shown in (25):

$$Mh(q)\left((\alpha+1)e_\vartheta + \beta e_\vartheta^{q/p}\right) - Vh(q,\dot{q})e_\vartheta + f(x) = Bh(q)\tau \tag{25}$$

The control law is obtained as shown in (26):

$$\tau = Bh^{-1}(C_v e_\vartheta + f(x) + \beta Mh(q)e_\vartheta^{q/p}) \tag{26}$$

Such that (27):

$$C_v = Mh(q)(\alpha+1) - Vh(q,\dot{q}) \tag{27}$$

### 4.2. Control in presence of disturbances

In this case, the disturbances are considered; therefore the neural network controller is introduced. The equation (19) is defined as shown in (28):

$$-Mh(q)\dot{e}_\vartheta - Vh(q,\dot{q})e_\vartheta + f(x) - \tau_d = Bh(q)\tau \tag{28}$$

The control law designed in equation (26) can be rewritten as shown in (29):

$$\tau = Bh^{-1}(C_v e_\vartheta + \hat{f}(x) + \beta Mh(q)e_\vartheta^{q/p}) \tag{29}$$

when $\hat{f}(x)$ is the output of RBF network. $\hat{f}(x)$ approximates $f(x)$

RBF network can be adopted to approximate *f(x).* The desired algorithm of RBF network is (30-32):

$$\varphi_i = g(x - c_i^2)/b_i^2 \quad i=1,2,\ldots\ldots..n \tag{30}$$

$$y = W^{*T}\varphi(x) \tag{31}$$

$$f(x) = W^{*T}\varphi(x) + \varepsilon \tag{32}$$

*x* is the input state of network, $\varphi(x) = [\ \varphi_1\ \varphi_2 .... \varphi_n]^T$, ε is the approximation error of network. *W** is the weight vector of desired RBF network. Replacing (29) in (28) and as shown in (33):

$$Mh(q)\dot{e}_\vartheta = -(Vh(q,\dot{q})+C_v)e_\vartheta - \beta_3 Mh(q)e_\vartheta{}^{q/p} + \mu_0 \tag{33}$$

when: $\mu_0 = \tilde{f}(x) - \tau_d$ , and $\tilde{f}(x) = f(x) - \hat{f}(x)$
The output of the network is giving as shown in (34):

$$f(x) = \hat{W}^T \varphi(x) \tag{34}$$

Selecting: $\hat{W} = W^* - \hat{W}$ , $\|W^*\| \le W_{\max}$
Therefore (35):

$$\mu_0 = \tilde{f}(x) - \tau_d = \hat{w}^T \varphi(x) + \varepsilon - \tau_d \tag{35}$$

The control law designed in equation (29) can rewrite as shown in (36):

$$\tau = Bh^{-1}\tau = Bh^{-1}(C_v e_\vartheta + \hat{f}(x) + \beta Mh(q) e_\vartheta{}^{\frac{q}{p}}) - \xi \tag{36}$$

where ξ is the robust element introduced to eliminate the network approximation error ε and the disturbances $\tau_d$ Replacing the equation (35) in (33) and shown in (37):

$$Mh(q)\dot{e}_\vartheta = -(Vh(q,\dot{q})+C_v)e_\vartheta - \beta Mh(q)e_\vartheta{}^{q/p} + \hat{W}^T \varphi(x) + \varepsilon - \tau_d + \xi \tag{37}$$

Putting (38):

$$\mu_1 = \hat{W}^t \varphi(x) + \varepsilon - \tau_d + \xi \tag{38}$$

Replacing (36) in (35) and as shown in (39):

$$Mh(q)\dot{e}_\vartheta = -(Vh(q,\dot{q})+C_v)e_\vartheta - \beta Mh(q)e_\vartheta{}^{q/p} + \mu_1 \tag{39}$$

with: $Vh(q,\dot{q}) + C_v = Mh(q)(\alpha+1)$

The robust element ξ is designed as shown in (40):

$$\xi = -(\varepsilon_N + b_d)\,\text{sign}(e_\vartheta) \tag{40}$$

Where: $\|\varepsilon\| \le \varepsilon_N$ , $\|\tau_d\| \le b_d$

The candidate function of lyapunov is selected as shown in (41):

$$\rho = \frac{1}{2} e_\vartheta{}^T Mh(q) e_\vartheta + \frac{1}{2} tr(\hat{W}^T \mathrm{F}_w{}^{-1} \hat{W}) \tag{41}$$

The derivative of the lyapunov function is defined as shown in (42):

$$\dot{\rho} = e_\vartheta{}^T Mh(q)\dot{e}_\vartheta + \frac{1}{2} e_\vartheta{}^T \dot{M}h(q) e_\vartheta + tr(\tilde{W}^T \mathrm{F}_w^{-1} \tilde{W}) \tag{42}$$

From equation (37) to be (43):

$$\dot{\rho} = -e_\vartheta{}^T Mh(q)(\alpha+1)e_\vartheta - e_\vartheta{}^T \beta Mh(q) e_\vartheta{}^{q/p} + tr\tilde{W}^T \left(\mathrm{F}_w^{-1}\tilde{W} + \varphi(x) e_\vartheta{}^T\right) + e_\vartheta{}^T (\varepsilon - \tau_d + \xi) \tag{43}$$

Select: $\tilde{W} = F_w \varphi(x) e_\vartheta^T$

The adaptive rule of network is (44):

$$\hat{W} = F_w \varphi(x) e_\vartheta^T \tag{44}$$

Therefore (45):

$$\dot{\rho} = -e_\vartheta^T Mh(q)(\alpha+1)e_\vartheta - e_\vartheta^T \beta Mh(q) e_\vartheta^{q/p} + e_\vartheta^T(\varepsilon - \tau_d + \xi) \tag{45}$$

Considering the term (46):

$$e_\vartheta^T(\varepsilon - \tau_d + \xi) = e_\vartheta^T(\varepsilon - \tau_d) + e_\vartheta^T \xi = e_\vartheta^T(\varepsilon - \tau_d) - \|e_\vartheta\|(\varepsilon_N + b_d) \le 0 \tag{46}$$

Such that matrix *Mh* define positive, α and β are positives, *p* and *q* are positives odd integers (0<*q*/*p*<1); therefore: $\dot{\rho} \le 0$

## 5. Simulations and Results

In this section the simulation using Matlab/Simulink is applied on the dynamic mobile robot system. Firstly, the disturbances are excluded, secondly the disturbances are injected, and finally the RBF neural network is introduced to estimate the system nonlinear function and deal the disturbances.

Let us consider: $v_r$ = 1 m/s, $\omega_r$ = 1 rad/s.

The disturbances $\tau_d$=[0.1.sin(t) 0.1.cos(t)].

α=1, β=2, *p*=7, *q*=5.$\varepsilon_N$=0.2, $b_d$=0.1.

The neural network is chosen with seven hidden, the initial weight matrix is selected as 0.1. b=10. The matrices values of the dynamic model are taken from [25]. Figures 3 and 4 show that actual forward and angular velocities of the control proposed could keep up with the desired ones in absence of distrubances, but the tracking is dgraded when the distrubances is appeared, In the case of RBF control the forward and angular velocities could keep up the desired ones in presence of disturbances.

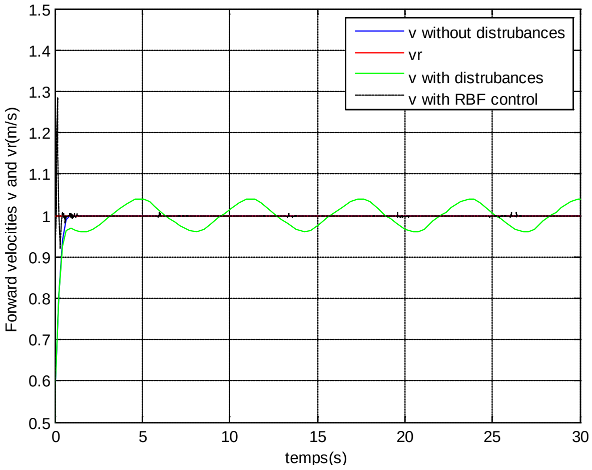

Figure 3. Forward velocities

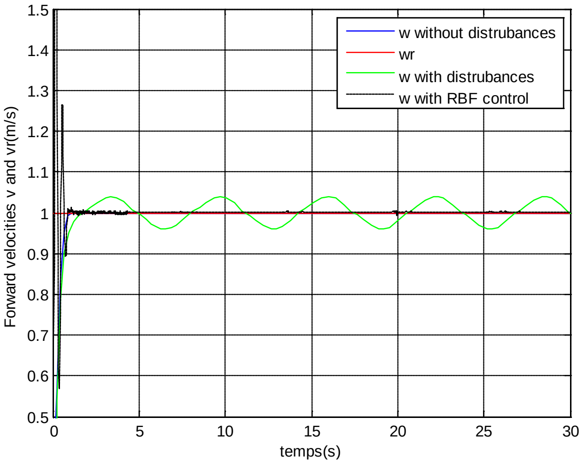

Figure 4. Angular velocities

The figure 5 and 6 show the control torques obtained in presence and absence of distrubances, the torques obtained in the first case are very smooth, when compared with the torques in the second case. In second part of simulations, the sinusoidal reference is considered: $v_r$ = cos (2Π.t/5), $\omega_r$ = sin (2Π.t/5). The disturbances $\tau_d$=[0.6.sin(t) 0.6.cos(t)]. In Figure 7 is non-linear and estimate function.

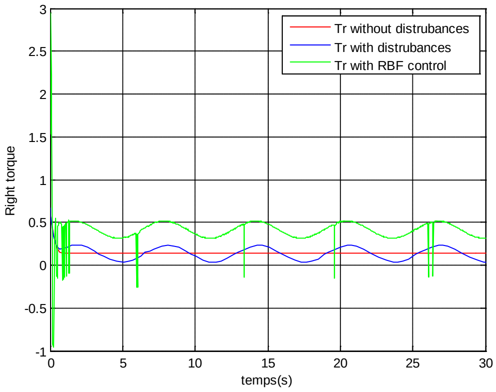

Figure 5. Input right torque

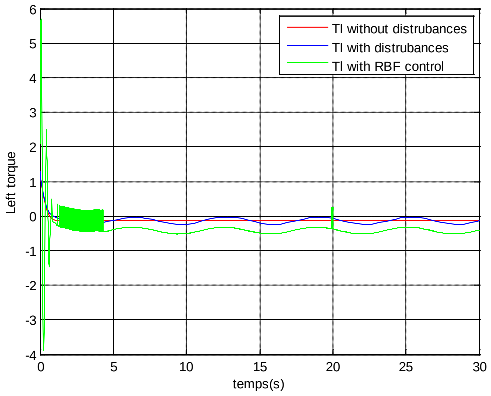

Figure 6. Input left torque

Figures 8 and 9 show that actual linear and angular velocities of the proposed controller could keep up with the desired ones in finite time when the distrubances is excluded, but the tracking is degraded when the distrubances is appeared. In the case of RBF control the linear and angular velocities could keep up the desired ones in presence of disturbances. In Figure 10 is Input right torque. In Figure 11 is input left torwue. Figure 12 show the estimated function of the non-linear function.

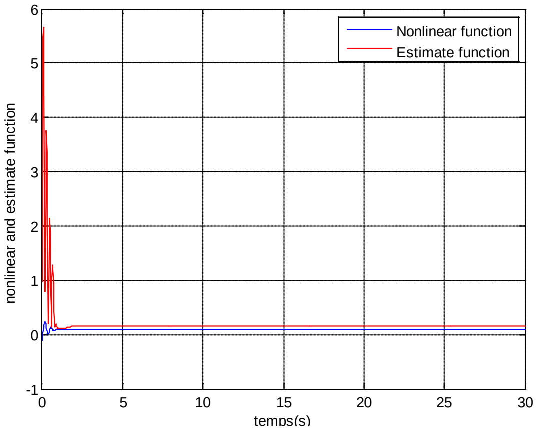

Figure 7. Non-linear and estimate function

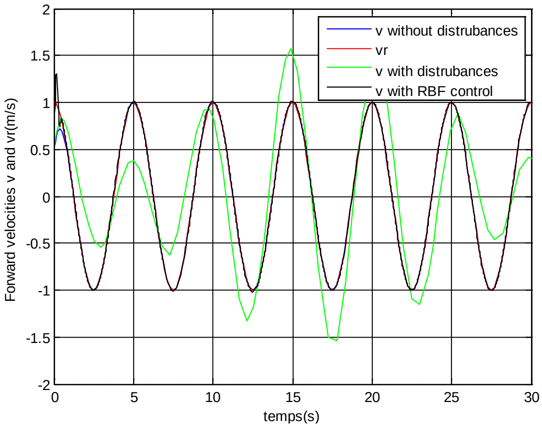

Figure 8. Forward velocities

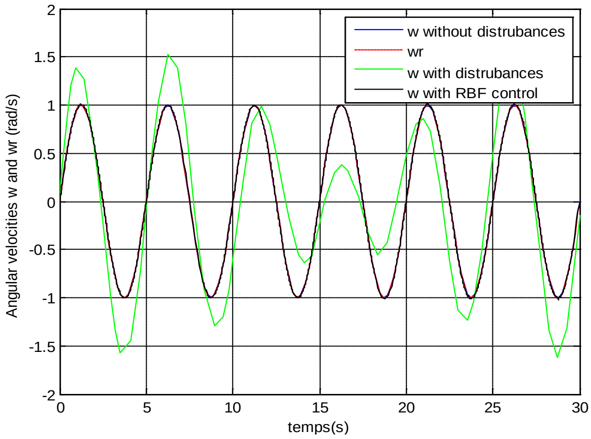

Figure 9. Angular velocities

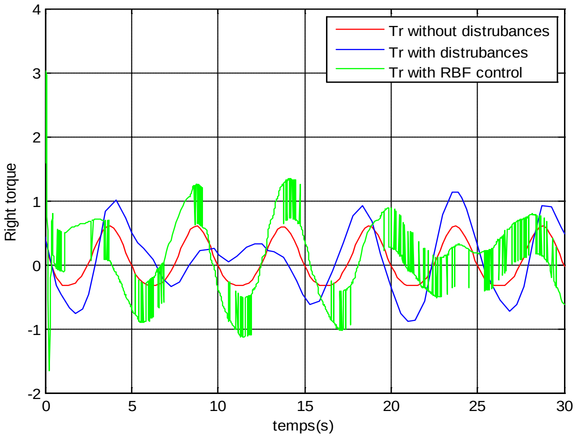

Figure 10. Input right torque

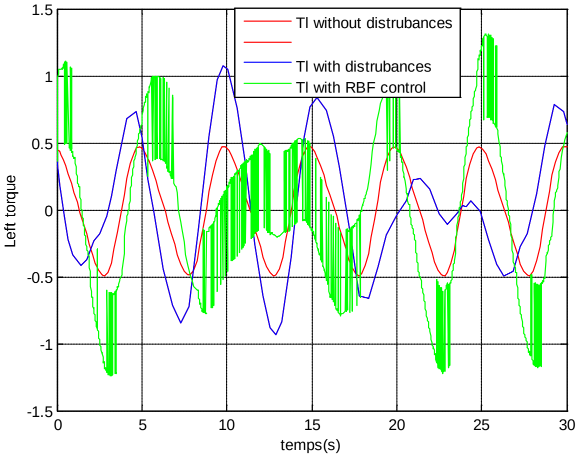

Figure 11. Input left torque

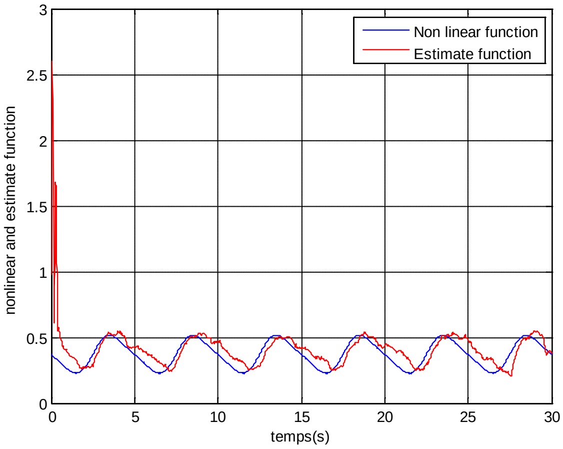

Figure 12. Non-linear and estimate function

## 6. CONCLUSION

In this paper a RBF-GFSM control is proposed to ensure the dynamic stability of mobile robot. The GFSM control is used in order to make the system converges to the reference in a finite time. The RBF controller stabilizes the velocities errors, deal the disturbances and approximate the system nonlinear function. Simulations results have demonstrated that the RBF-GFSM is efficiency and gives the best performances in comparison with GFSM in the case where the disturbances are introduced.

## REFERENCES

[1] C. Samson, Velocity and torque feedback control of a nonholonomic cart, in Lecture Notes in Control and Information Science, C. Canudas de Wit, Ed. Berlin, Germany: Springer-Verlag, 1991, pp. 125–151.

[2] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, Nonlinear control design for mobile robots, *in Recent Trends in Mobile Robots, Y. F. Zheng, Ed. Singapore: World Scientific, 1993, pp. 121–156.*

[3] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, Control and stabilization of nonholonomic dynamic systems, *IEEE Trans. Automat. Contr., vol. 37, pp. 1746–1757, 1992.*

[4] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, A stable tracking control method for an autonomous mobile robot, *in Proc. IEEE Int. Conf. Robot. Automat., 1990, pp. 384–389.*

[5] Fukao, T., Nakagawa, H. and Adachi, N, Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation.2000* 16 (5), *609–615.*

[6] Hwang, C. L. (2004). A novel Takagi-Sugeno-based robust adaptive fuzzy sliding-mode controller. *IEEE Transactions on Fuzzy Systems.* 12 (5), *676–687.*

[7] Mohareri, O., Dhaouadi, R., Shirazi, M.M. Intelligent neural network based controllers for path tracking of wheeled mobile robots: A comparative analysis. *IEEE International Workshop on Robotic and Sensors Environments,2010 pp. 1-6.*

[8] Kozlowski, K. and Majchrzak, J. A backstepping approach to control a nonholonomic mobile robot. *In Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4.*2002. pp. 3972-3977.*

[9] Nganga-Kouya, D. and Okou, F.A. Adaptive backstepping control of a wheeled mobile robot. *17th Mediterranean Conference on Control and Automation, 2009. pp. 85-91.*

[10] Song, K.T., Lee, C.H. and Teng, C.C., Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *IEEE Transactions on Control Systems Technology, 2001 pp. 305-318.*

[11] Yang, J.M. and Kim, J.H. Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots. *IEEE Transactions on Robotics and Automation.*1999 15 (3), *578–587.*

[12] R. Fierro and F.L. lewis. Control of nonholonomic mobile robot using neural network, *IEEE Trans.Neural Network* 9(4).*589-600(1998).*

[13] Jin Oh Jang. Adaptive neuro-fuzzy network control for a mobile robot. J *Intell Robot Syst (2011).* 62, *567-586.*

[14] Park B. S., Yoo S. J., Park J. B., and Choi Y. H. Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty, *IEEE Trans. Control Syst. Technol., Vol. 17, no. 1, pp. 207– 214, Jan 2009.*

[15] Ou. M.Y, Li. S.H, Wang. C.L, Finite-time tracking control for nonholonomic mobile robots based on visual servoing, Asian J. *Contr.* (3) *(2014) 1–13.*

[16] Wu. Y.Q, Wang. B, Zong. G.D, Finite-time tracking controller design for nonholonomic systems with extended chained form, *IEEE Trans. Circ. Syst. 52* (11) *(2005) 798–802.*

[17] Du. H.B, Li. S.H, Qian. C.J, Finite-time attitude tracking control of spacecraft with application to attitude synchronization, *IEEE Trans. Autom. Contr. 56* (11) *(2011) 2711–2717.*

[18] Li. S.H, Ding. S.H, Li. Q., Global set stabilization of the spacecraft attitude using finite-time control technique, *Int. J. Contr. 82* (5) *(2009) 822–836.*

[19] Khoo, S., Xie, L. and Man, Z. Robust finite-time consensus tracking algorithm for multirobot systems. *IEEE/ASME Transactions on mechatronics*, (2009) *14* (2), *219-228*.

[20] Ying, J., Khoo, S., Man, Z. and Yu, X. Finite-time stability and instability of stochastic nonlinear systems. *Automatica*, (2011) *47* (12), *2761-2677*.

[21] Polyakov, A. Nonlinear feedback design for fixed-time stabilization of linear control systems, *IEEE Transactions on Automatic Control*, *(2012) 57* (8), *2106–2110*.

[22] Khoo, S., Ying, J., Man, Z. and Yu, X. Finite-time stabilization of stochastic nonlinear systems in strict-feedback form. *Automatica*, *(2013) 49* (5), *1403-1410*.

[23] R. Fierro and F. L. Lewis, Control of a Nonholonomic Mobile Robot Using Neural Networks, *in IEEE Transactions on Neural Networks, vol. 9, pp. 589-600, July 1998.*

[24] Park J, Sandberg IW. Universal approximation using radial-basis-function networks. Neural computation, *1991,3: 246 257.*

[25] Duc Do, K., Zhong-Ping J., Pan, J. A global output-feedback controller for simultaneous tracking and stabilization of unicycle-type mobile robots, *IEEE Trans. Automat. Contr., V30, N3, pp. 589–594, 06/04*.