Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 7(2): 121–134.
DOI: 10.3934/electreng.2023007
Received: 24 October 2022
Revised: 01 March 2023
Accepted: 10 March 2023
Published: 29 March 2023

http://www.aimspress.com/journal/ElectrEng

Research article

Adaptive PID sliding mode control based on new Quasi-sliding mode and radial basis function neural network for Omni-directional mobile robot

Thanh Tung Pham[1,*] and Chi-Ngon Nguyen[2]

[1] Vinh Long University of Technology Education, Vietnam
[2] Can Tho University, Vietnam

*** Correspondence:** Email: tungpt@vlute.edu.vn; Tel: +906-295-268.

Abstract: This article designs a PID sliding mode controller based on new Quasi-sliding mode (PID-SMC-NQ) and radial basis function neural network (RBFNN) for Omni-directional mobile robot. This is holonomic vehicles that can perform translational and rotational motions independently and simultaneously. The PID-SMC is designed to ensure that the robot's actual trajectory follows the desired in a finite time with the error converges to zero. To decrease chattering phenomena around the sliding surface, in the controller robust term, this paper uses the *tanh* (hyperbolic tangent) function, so called the new Quasi-sliding mode function, instead of the switch function. The RBFNN is used to approximate the nonlinear component in the PID-SMC-NQ controller. The RBFNN is considered as an adaptive controller. The weights of the network are trained online due to the feedback from output signals of the robot using the Gradient Descent algorithm. The stability of the system is proven by Lyapunov's theory. Simulation results in MATLAB/Simulink show the effectiveness of the proposed controller, the actual response of the robot converges to the reference with the rising time reaches 307.711 ms, 364.192 ms in the x-coordinate in the two-dimensional movement of the robot, the steady-state error is 0.0018 m and 0.00007 m, the overshoot is 0.13% and 0.1% in the y-coordinate, and the chattering phenomena is reduced.

Keywords: PID sliding mode control; new Quasi-sliding mode; radial basis function neural network; gradient descent; Omni-directional mobile robot; MATLAB/Simulink

1. Introduction

Omni-directional mobile robots (OMRs) have been used in a wide range of fields, in the narrow spaces requiring high mobility in factories and hospitals [1], personal assistance rehabilitation, industrial applications, service robots, hobby and competition [2]. The main benefit of an Omni-directional motion system is that it provides three degrees of freedom (DOFs) in a ground plane, allowing displacements in any direction while changing its orientation [3].

Many different approaches in the trajectory tracking control design for such models of Omni-directional mobile robots have been proposed, including a fuzzy controller was designed in [2]. After 2.23 s, the robot reached the target and stayed there the rest of the time, the maximum tracking error of the proposed approach was less than 1 mm. To reduce the tracking error of the robot, the PI controllers, as presented in [4]. A circle was assigned as the path; the motion starts from A (0,1) in world coordinates the motion. The commanded signal and its response were very close and the difference was slightly appearable. Beside, to performance evaluation of the global trajectory tracking control problem, the PD+ approach was developed in [5]. The value of ISI (Integral of the Squared Input) index of the controllers was 8705235.5, the RMSE (Root Mean Square Error) index for e_q and $\dot{e}_q$ was 46.41 and 6.56, respectively. To perfect in the aspects of stability control and theoretical analysis, the control of adaptive gain radial basis neural network (RBF) on PI dynamic sliding mode dynamic controller was proposed in [6]. In this experiment, the control of the proposed method can make the sine like curve formation motion consistent with the desired trajectory, and the tracking curve was smooth. However, the disturbance and tracking error in the process of robot motion were occur because of the great influence of friction and communication delay. To reduce the tracking error of the robot, an adaptive nonlinear control was designed and simulated in [7]. Simulation results showed that the circle trajectory of the robot reached the desired in 0.5 sec without overshoot, the steady-state error was negligible. To solve the constrained control problems and parameter uncertainties, an adaptive model predictive control (MPC) scheme with friction compensation was developed in [1]. Experimental results with the eight-shape trajectory showed that the IAE_{xy} (Integral of Absolute Error) and IAE_{θ} of model-based adaptive MPC were, respectively, 1.517 m and 1.2907 rad. MAE_{xy} (Maximum Absolute Error) and MAE_{θ} of model-based adaptive MPC were 0.0997 m and 0.0655 rad, respectively. Experimental tests have demonstrated that the proposed control method can cope with parameter uncertainties, with comparison against model-based MPC However, the control scheme depends on the dynamic model, and parameter estimation errors exist.

To reduce the tracking error and the effect of disturbance, the present study proposes a PID sliding mode controller based on new Quasi-sliding mode (PID-SMC-NQ) and the radial basis function neural network (RBFNN) to control the trajectory tracking for Omni-directional mobile robot. The PID-SMC-NQ is designed to ensure that the robot's actual trajectory follows the desired and reduce the chattering phenomena around the sliding surface. The RBFNN is used to approximate the nonlinear component ($\boldsymbol{A}_w$ matrix) in the PID-SMC-NQ controller and is considered as an adaptive controller. The weights of the RBFNN are trained online due to the feedback from output signals of the robot using the Gradient Descent algorithm.

This paper is organized in five sections: Section 2 presents the mathematical model of the robot. The PID sliding mode control based on new Quasi-sliding mode and the RBFNN is presented in Section 3. The simulation and evaluation results are presented in Section 4, and Section 5 contains the conclusions.

2. Mathematical model of the Omni-directional mobile robot

The model of the Omni-directional mobile robot is presented as Figure 1 [8]. It is assumed 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 on the center of gravity for the mobile robot.

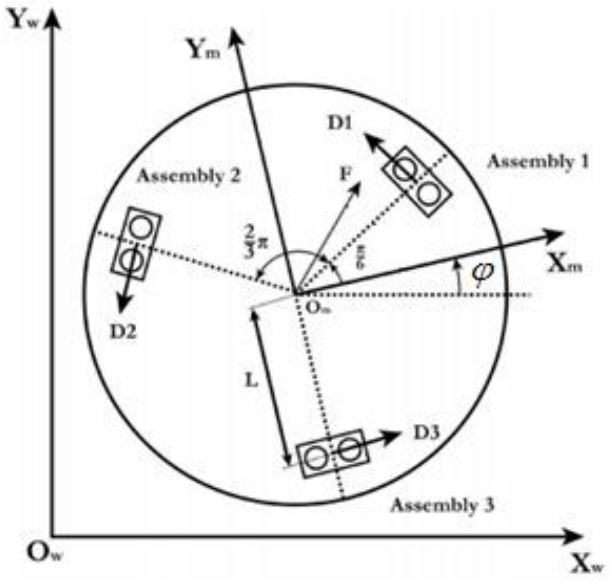


Figure 1. Model of the Omni-directional mobile robot [8].

The dynamic model of Omni-directional mobile robot can be represented as (1) [8].

$$\begin{bmatrix} \ddot{x}_w \\ \ddot{y}_w \\ \ddot{\varphi} \end{bmatrix} = \begin{bmatrix} a_1 & -a_2\dot{\varphi} & 0 \\ a_2\dot{\varphi} & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix} \begin{bmatrix} \dot{x}_w \\ \dot{y}_w \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\varphi \\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\varphi \\ b_2 & b_2 & b_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} + \begin{bmatrix} D_{fx} \\ D_{fy} \\ D_{f\varphi} \end{bmatrix} = \boldsymbol{A}_w\boldsymbol{X} + \boldsymbol{B}_w\boldsymbol{U} + \boldsymbol{D}_f \quad (1)$$

where $\boldsymbol{X} = \begin{bmatrix} \dot{x}_w & \dot{y}_w & \dot{\varphi} \end{bmatrix}^T$ are the state vector, $\boldsymbol{U} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$ are the driving input torque and $\boldsymbol{D}_f = \begin{bmatrix} D_{fx} & D_{fy} & D_{f\varphi} \end{bmatrix}^T$ are unknown system disturbances. $\boldsymbol{A}_w$ and $\boldsymbol{B}_w$ are calculated based on robot's parameters as follows:

$$\boldsymbol{A}_w = \begin{bmatrix} a_1 & -a_2\dot{\varphi} & 0 \\ a_2\dot{\varphi} & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix}, \boldsymbol{B}_w = \begin{bmatrix} b_1\gamma_1 & b_1\gamma_2 & 2b_1\cos\varphi \\ b_1\gamma_3 & b_1\gamma_4 & 2b_1\sin\varphi \\ b_2 & b_2 & b_2 \end{bmatrix}, \quad a_2 = 1 - a_2' = \frac{3I_w}{(3I_w + 2Mr^2)}$$

$$a_1 = \frac{-3c}{(3I_w + 2Mr^2)};\ a_2' = \frac{2Mr^2}{(3I_w + 2Mr^2)};\ a_3 = \frac{-3cL^2}{(3I_wL^2 + I_vr^2)};\ b_1 = \frac{kr}{(3I_w + 2Mr^2)};\ b_2 = \frac{krL}{(3I_w + I_vr^2)}$$

$$\gamma_1 = -\sqrt{3}\sin\varphi - \cos\varphi, \gamma_2 = \sqrt{3}\sin\varphi - \cos\varphi, \gamma_3 = \sqrt{3}\cos\varphi - \sin\varphi, \gamma_4 = -\sqrt{3}\cos\varphi - \sin\varphi$$

where M is the mass of the robot; L is the distance from any wheel and the center of gravity of the robot; k is the driving gain factor; r is the radius of each wheel of robot; c is the viscid resistance factor of the wheel; I_w is the moment of inertia of the wheel of robot around the driving shaft and I_v is the moment of inertia for the robot.

3. Design of an adaptive PID sliding mode control based on new Quasi-sliding mode and the RBFNN

3.1. PID sliding mode control design based on new Quasi-sliding mode

The structure diagram of the PID-SMC-NQ is presented as Figure 2.

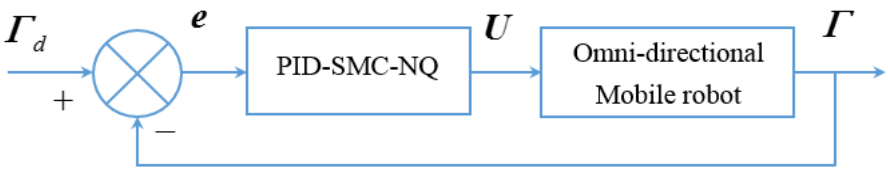


Figure 2. Structure diagram of the PID-SMC-NQ.

where $\boldsymbol{\Gamma}_d = \begin{bmatrix} x_d & y_d & \varphi_d \end{bmatrix}^T$ are the desired trajectory and $\boldsymbol{\Gamma} = \begin{bmatrix} x_w & y_w & \varphi \end{bmatrix}^T$ are the actual trajectory of the robot.

In this research, the PID-SMC-NQ is designed to control the actual trajectory of the robot tracks the desired in a finite time and reduce the chattering phenomena around the sliding surface.

The error between the actual trajectory and the desired of the robot is defined as (2):

$$\boldsymbol{e} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathrm{d}} \tag{2}$$

Taking the first and second order derivative of (2), we get (3) and (4):

$$\dot{\boldsymbol{e}} = \dot{\boldsymbol{\Gamma}} - \dot{\boldsymbol{\Gamma}}_{\mathrm{d}} \tag{3}$$

$$\ddot{\boldsymbol{e}} = \ddot{\boldsymbol{\Gamma}} - \ddot{\boldsymbol{\Gamma}}_{\mathrm{d}} \tag{4}$$

The PID sliding surface for the sliding mode control can be indicated using the following equation [9]:

$$\boldsymbol{S} = \boldsymbol{K}_P \boldsymbol{e} + \boldsymbol{K}_I \int \boldsymbol{e}(\tau) d\tau + \boldsymbol{K}_D \dot{\boldsymbol{e}} \tag{5}$$

where $\boldsymbol{K}_P = diag\left(K_{P1}, K_{P2}, K_{P3}\right)$, $\boldsymbol{K}_I = diag\left(K_{I1}, K_{I2}, K_{I3}\right)$, $\boldsymbol{K}_D = diag\left(K_{D1}, K_{D2}, K_{D3}\right)$ are designed positive constants.

Taking the derivative of (5), we get (6):

$$\dot{\boldsymbol{S}} = \boldsymbol{K}_P \dot{\boldsymbol{e}} + \boldsymbol{K}_I \boldsymbol{e} + \boldsymbol{K}_D \ddot{\boldsymbol{e}} \tag{6}$$

Substituting (4) into (6), we get (7):

$$\dot{\boldsymbol{S}} = \boldsymbol{K}_P\dot{\boldsymbol{e}} + \boldsymbol{K}_I\boldsymbol{e} + \boldsymbol{K}_D\left(\boldsymbol{A}_w\boldsymbol{X} + \boldsymbol{B}_w\boldsymbol{U} + \boldsymbol{D}_f - \ddot{\boldsymbol{\Gamma}}_d\right) \tag{7}$$

To decrease chattering, in the controller robust term, we use the *tanh* function, so called the new Quasi-sliding mode function, instead of the switch function [10], i.e.,

$$\dot{\boldsymbol{S}} = -\boldsymbol{\eta}\tanh\left(\boldsymbol{S}/\boldsymbol{\varepsilon}\right) \tag{8}$$

where $\boldsymbol{\eta} = diag\left(\eta_1, \eta_2, \eta_3\right)$, $\boldsymbol{\varepsilon} = diag\left(\varepsilon_1, \varepsilon_2, \varepsilon_3\right)$ are symmetric positive definite.

We get the PID-SMC-NQ law as (9):

$$\boldsymbol{U}_{PID-SMC-NQ} = -\left(\boldsymbol{K}_D\boldsymbol{B}_w\right)^{-1}\left(\boldsymbol{K}_P\dot{\boldsymbol{e}} + \boldsymbol{K}_I\boldsymbol{e} + \boldsymbol{K}_D\left(\boldsymbol{A}_w\boldsymbol{X} - \ddot{\boldsymbol{\Gamma}}_d\right) + \boldsymbol{\eta}\tanh\left(\boldsymbol{S}/\boldsymbol{\varepsilon}\right)\right) \tag{9}$$

As defined in (1), the determinant of $\boldsymbol{B}_w$ as (10):

$$det\left(\boldsymbol{B}_w\right) = 6\sqrt{3}b_1^2b_2 \neq 0 \tag{10}$$

Because of determinant of the matrix $\boldsymbol{B}_w$ is nonzero, so the inverse of matrix $\boldsymbol{B}_w$ exists, hence the PID-SMC-NQ law for the robot is presented as (9) exists and ensures that the actual trajectory of the robot converges to the desired trajectory in a finite time and reduces the chattering around the sliding surface.

3.2. An adaptive PID-SMC-NQ controller based on the RBFNN Approximation

The RBFNN is a single hidden layer neural network [11] and can be consider as a mapping: $R^r \to R^s$, which embraces three different layers: an input layer, a hidden layer and an output layer [12–14].

Input layer: training and testing samples.

Hidden layer: the number of hidden layer nodes depends on the requirement. Radial basis function, typically Gaussian function, as the activation function of hidden layer to transform the input information into space mapping.

Output layer: respond to input mode. The action function of the output layer neurons is a linear function. And take the weighed sum of the output information of the hidden layer as the output of the whole neural network.

The RBFNN has the advantages of simple structure design, easy training, fast convergence, can effectively fit any nonlinear function and is not easy to fall into the local optimal solution [11,14]. The RBFNN has many uses, including function approximation, classification, and system control. They have the advantage of fast learning speed and are able to avoid the problem of local minimum [14,15].

The structure [5-7-1] of the RBFNN is used in this article to approximate the functions $a_{i|i=1,2,3}$ in the matrix $\boldsymbol{A}_w$ of the control law in (9) is illustrated in Figure 3 [13,15].

$\boldsymbol{A}_w$ 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 RBFNN uses Gradient Descent algorithms to online update the weight values. Each RBF neural network contains 7 Gaussian functions that can be described as (11):

$$h_{ij} = \exp\left(-\frac{\left\| \boldsymbol{x}_i - \boldsymbol{c}_{ij} \right\|^2}{2\boldsymbol{b}_{ij}^2} \right)\Bigg|_{i=\overline{1,3};\, j=\overline{1,7}} \tag{11}$$

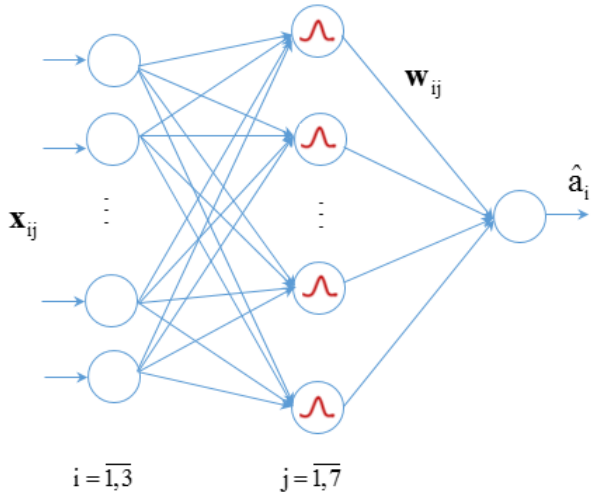


Figure 3. RBF neural network structure.

where $\boldsymbol{c}_{ij}$ represents the coordinate value of center point of the Gaussian function of neural net j for the ith input, $\boldsymbol{b}_{ij}$ represents the width value of Gaussian function for neural net j for the ith input, and:

$$\boldsymbol{x}_i = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} e(1) & \dot{e}(1) & \Gamma_d(1) & \dot{\Gamma}_d(1) & \ddot{\Gamma}_d(1) \\ e(2) & \dot{e}(2) & \Gamma_d(2) & \dot{\Gamma}_d(2) & \ddot{\Gamma}_d(2) \\ e(3) & \dot{e}(3) & \Gamma_d(3) & \dot{\Gamma}_d(3) & \ddot{\Gamma}_d(3) \end{bmatrix} \tag{12}$$

$$h_{ij|i=1,2,3} = \begin{bmatrix} h_{i1} & h_{i2} & h_{i3} & h_{i4} & h_{i5} & h_{i6} & h_{i7} \end{bmatrix} \tag{13}$$

$$w_{ij|i=1,2,3} = \begin{bmatrix} w_{i1} & w_{i2} & w_{i3} & w_{i4} & w_{i5} & w_{i6} & w_{i7} \end{bmatrix} \tag{14}$$

The outputs of the RBFNN are given by (15):

$$\hat{a}_i = \boldsymbol{w}_{ij}^T \boldsymbol{h}_{ij} \tag{15}$$

The performance index function of the RBFNN as (16):

$$E_i(t) = \frac{1}{2}\left(a_i(t) - \hat{a}_i(t)\right)^2;\ i = 1,2,3 \tag{16}$$

According to Gradient Descent method, the weight values can be updated as (17) and (18):

$$\Delta w_j(t) = -\mu \frac{\partial E}{\partial w_j} = \mu\left(a_i(t) - \hat{a}_i(t)\right) h_j \tag{17}$$

$$w_j(t) = w_j(t-1) + \Delta w_j(t) + \alpha\left(w_j(t-1) - w_j(t-2)\right) \tag{18}$$

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

Hence, the approximation matrix can be calculated as (19):

$$\hat{\boldsymbol{A}}_w = \begin{bmatrix} \boldsymbol{w}_{1j}^T \boldsymbol{h}_{1j} & -\boldsymbol{w}_{2j}^T \boldsymbol{h}_{2j} \dot{\varphi} & 0 \\ \boldsymbol{w}_{2j}^T \boldsymbol{h}_{2j} \dot{\varphi} & \boldsymbol{w}_{1j}^T \boldsymbol{h}_{1j} & 0 \\ 0 & 0 & \boldsymbol{w}_{3j}^T \boldsymbol{h}_{3j} \end{bmatrix} \tag{19}$$

Now, the PID-SMC-NQ law (9) is called the adaptive PID-SMC-NQ based on the RBFNN. So that, (9) can be rewritten as (20):

$$\boldsymbol{U}_{APID-SMC-NQ-RBF} = -\left(\boldsymbol{K}_D \boldsymbol{B}_w\right)^{-1} \left(\boldsymbol{K}_P \dot{\boldsymbol{e}} + \boldsymbol{K}_I \boldsymbol{e} + + \boldsymbol{K}_D \left(\hat{\boldsymbol{A}}_w \boldsymbol{X} - \ddot{\boldsymbol{\Gamma}}_d \right) + \boldsymbol{\eta} \tanh\left(\boldsymbol{S} / \boldsymbol{\varepsilon} \right) \right) \tag{20}$$

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 $\boldsymbol{e} = \boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\mathrm{d}}$ are changed. At that time, the RBFNN 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 (20), the proposed controller can adapt to the conditions of the robot.

To prove the stability, the Lyapunov function can be defined by (21):

$$\boldsymbol{V} = \frac{1}{2} \boldsymbol{S}^2 \tag{21}$$

Taking the derivative of (21), we get (22):

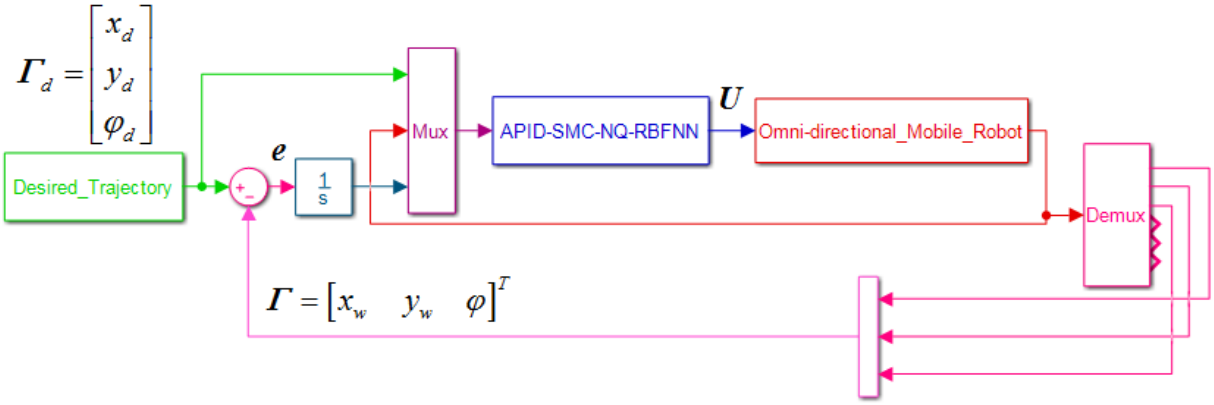
$$\dot{\boldsymbol{V}} = \boldsymbol{S}\dot{\boldsymbol{S}} = \boldsymbol{S} \left\{ \boldsymbol{K}_P \dot{\boldsymbol{e}} + \boldsymbol{K}_I \boldsymbol{e} + + \boldsymbol{K}_D \left(\hat{\boldsymbol{A}}_w \boldsymbol{X} + \boldsymbol{B}_w \boldsymbol{U} + \boldsymbol{D}_f - \ddot{\boldsymbol{\Gamma}}_d \right) \right\} = -\boldsymbol{\eta} \boldsymbol{S} \tanh\left(\boldsymbol{S} / \boldsymbol{\varepsilon} \right) \leq 0 \tag{22}$$

We can conclude that $\boldsymbol{S} \to \boldsymbol{0}$ at $t \to 0$ therefore, $\boldsymbol{e}, \dot{\boldsymbol{e}} \to \boldsymbol{0}$ at $t \to 0$.

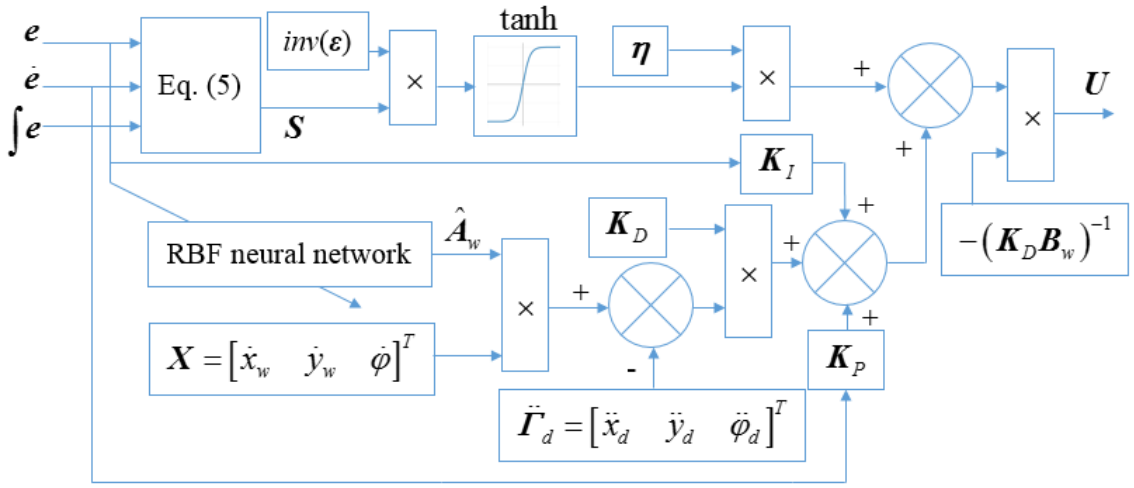
4. Simulation results and evaluations

The proposed controller simulation diagram for the robot in MATLAB/Simulink is presented as Figure 4a, Figure 4b presents detail diagram of the PID-SMC-NQ-RBF.

Model parameters are used for the simulation are given in Table 1. Table 2 presents the proposed controller parameters. The number of neurons in hidden layer is kept as 7 for all simulation cases.



(a) Simulation diagram of the PID-SMC-NQ-RBF for the robot in MATLAB/Simulink



(b) Detail diagram diagram of the PID-SMC-NQ-RBF

Figure 4. Simulation and detail diagram of the PID-SMC-NQ-RBF for the robot.

Table 1. Model Parameters of the Omni-directional mobile robot.

Parameters	Description	Value	Unit
I_v	Robot Moment of Inertia	11.25	kgm^2
M	Robot mass	9.4	kg
L	Distance from any wheel and the center of gravity of the robot	0.178	m
k	Driving Gain Factor	0.448	
c	Viscous Friction Factor	0.1889	kgm^2s^{-1}
I_w	Moment of Inertia of Wheel	0.02108	kgm^2
r	Radius of Wheel	0.0245	m

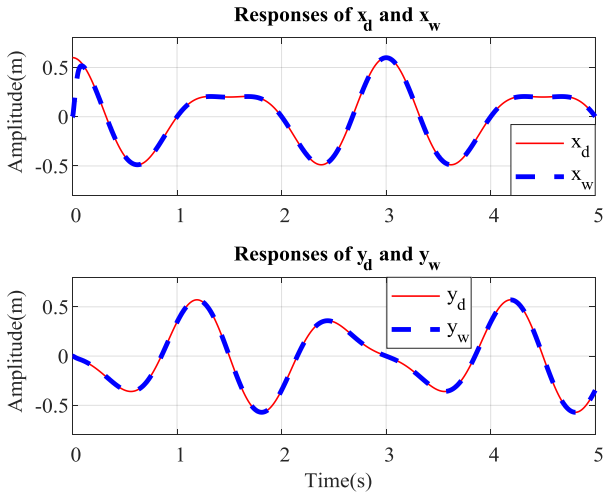


Figure 5. Response of x_d and x_w, y_d and y_w.

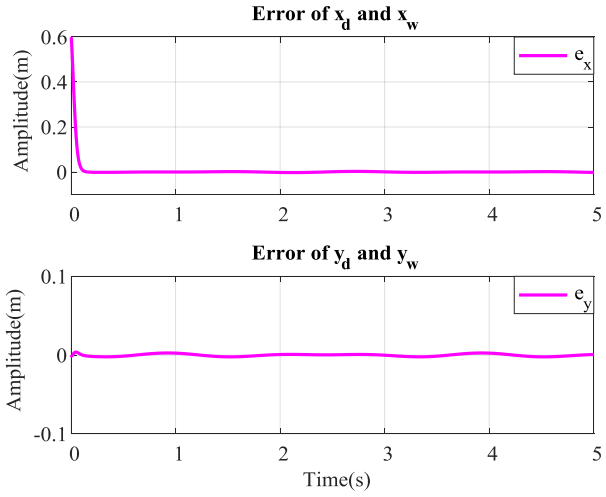


Figure 6. Tracking error of x_d and x_w, y_d and y_w.

The responses of x_d and x_w, y_d and y_w with the Hypocycloid trajectory of the PID-SMC-NQ-RBF controller are presented as Figure 5, in which we can see that the actual response of the x_w and y_w converges to the reference x_d and y_d with the rising time reached 307.711 ms, 364.192 ms, the steady-state error is 0.0018 m and 0.00007 m, which is shown in Figure 6, while the overshoot is 0.13% and 0.1%, respectively. These criteria are presented in Table 3 and compared with the GD-ASMC-RBF (Gradient-Adaptive Sliding Mode Control- Radial Basis Function) controller [16].

Table 2. The proposed controller parameters.

Parameters	Value
$\boldsymbol{K_P}$	diag(2; 2; 2)
$\boldsymbol{K_I}$	diag(0.02; 0.02; 0.02)
$\boldsymbol{K_D}$	diag(0.01; 0.01; 0.01)
$\boldsymbol{\eta}$	diag(25; 25; 25)
$\boldsymbol{\varepsilon}$	diag(0.5; 0.5; 0.5)
$\boldsymbol{c}$	$2*\begin{bmatrix} -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 \\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 \\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 \\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 \\ -1.5 & -1 & -0.5 & 0 & 0.5 & 1 & 1.5 \end{bmatrix}$
b	0.1
α	0.5

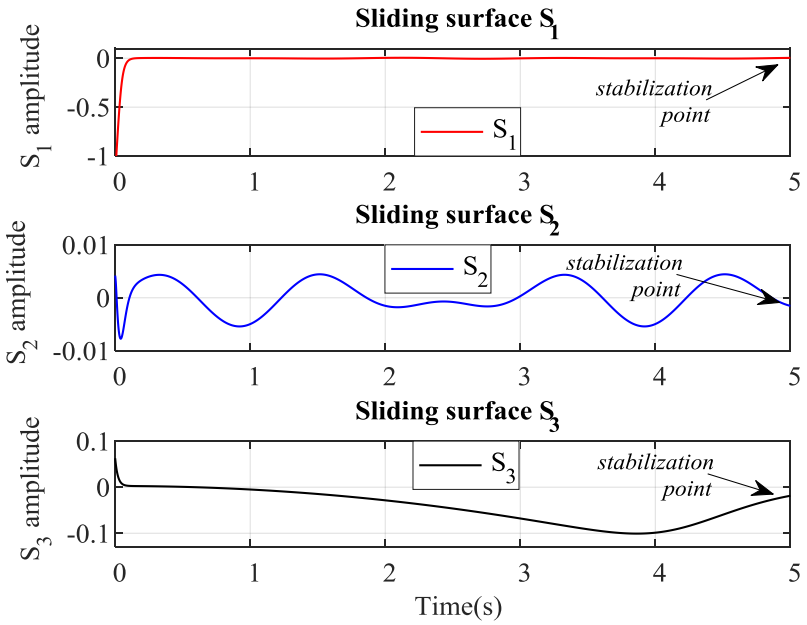


Figure 7. Sliding surface.

Table 3. The achieved quality criteria of the PID-SMC-NQ-RBF controller.

Quality criteria		Rising time (ms)	Overshoot (%)	Steady state error (m)
PID-SMC-NQ-RBF	x_w	307.711	0.13	0.0018
	y_w	364.192	0.1	0.00007
GD-ASMC-RBF	x_w	-	0.2	0.014
[16]	y_w	-	0.14	0.016

Figure 7 presents the sliding surface $\boldsymbol{S} = \begin{bmatrix} S_1 & S_2 & S_3 \end{bmatrix}^T$ of the proposed controller. This sliding surface at start-up according to the sliding coefficient value. Then, $\boldsymbol{S}$ rapidly reaches the convergence point (stabilization point) and keeps sliding around $\boldsymbol{S} = \boldsymbol{0}$.

Table 4 also provides various error performance measures for each response. The different error measures reported in the table are expressed as (23), (24), (25), (26), (27) and (28) [17].

Average Absolute Deviation:

$$AAD = \frac{1}{N}\sum_{t=1}^{N}\left|P_w(t) - P_d(t)\right| \tag{23}$$

Mean Square Error:

$$MSE = \frac{1}{N}\sum_{t=1}^{N}\left(P_w(t) - P_d(t)\right)^2 \tag{24}$$

Root Mean Square Error:

$$RMSE = \sqrt{MSE} \tag{25}$$

Mean Percentage Error:

$$MPE = \frac{1}{N}\sum_{t=1}^{N}\frac{\left(P_w(t) - P_d(t)\right)}{P_d(t)} \tag{26}$$

Mean Absolute Percentage Error:

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{\left(P_w(t) - P_d(t)\right)}{P_d(t)}\right| \tag{27}$$

Mean Relative Error:

$$MRE = \frac{100}{N}\sum_{t=1}^{N}\left|\frac{\left|P_w(t) - P_d(t)\right|}{P_d(t)}\right| \tag{28}$$

Table 4. Error performance measures.

Signals	x_w	y_w
AAD	3.6048e-07	1.4825e-07
MSE	6.4987e-10	1.0991e-10
RMSE	2.5493e-05	1.0484e-05
MPE	-2.5976e+09	4.2796e-07
MAPE	2.5976e+09	4.2796e-07
MRE	-2.5976e+11	-4.2796e-05

All the measures in Table 4 indicate that the actual trajectory in the robot's motion always follows the desired trajectory. The results clearly demonstrate that the Gradient Descent algorithm provides an accurate nonlinear predictive model.

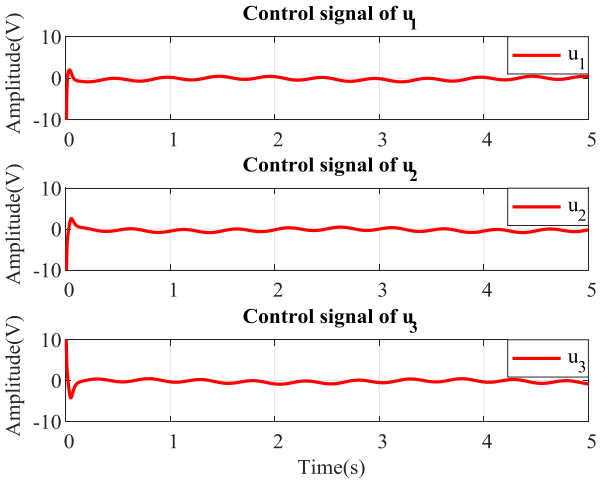


Figure 8. Control signal with the Hypocycloid trajectory.

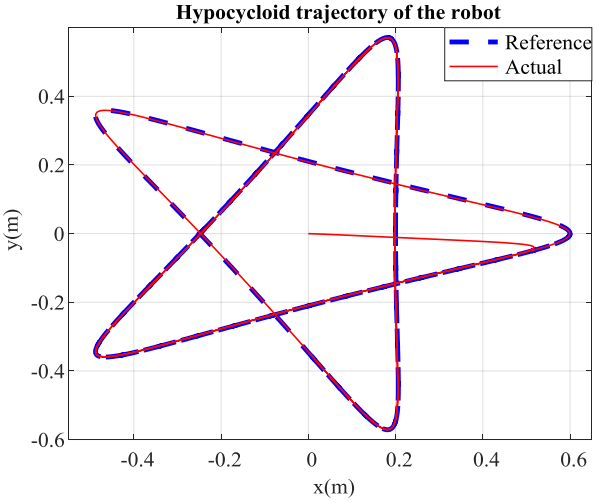


Figure 9. Robot trajectory for a reference path.

The control signal with the Hypocycloid trajectory illustrated in Figure 8 shows that the chattering phenomenon was reduced, with the amplitude converging to zero. This result demonstrates the effectiveness of the PID-SMC-NQ-RBF algorithm in controlling the robot.

The Hypocycloid trajectory response with the PID-SMC-NQ-RBF algorithm is presented as Figure 9. The actual trajectory of the robot tracks to the reference in a finite time with the error converges to zero.

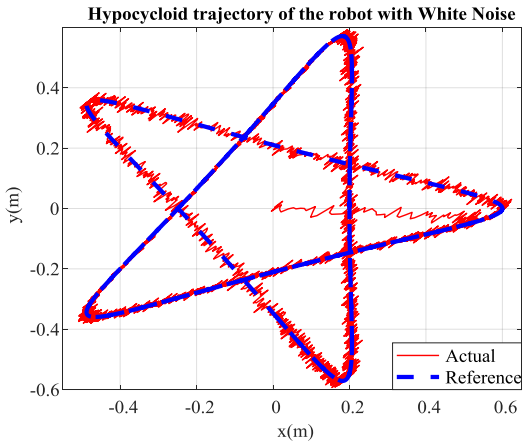


Figure 10. Robot trajectory with White Noise.

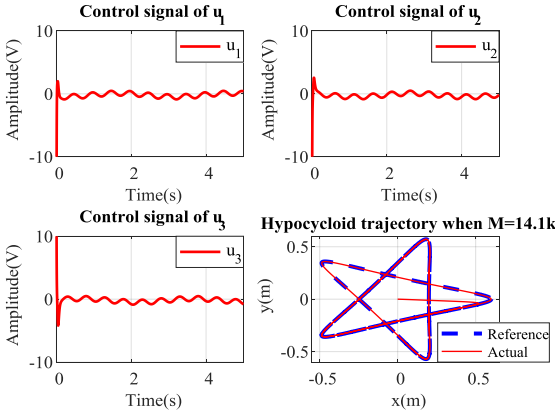


Figure 11. Robot trajectory when M is increased by 50%.

Figure 10 and Figure 11 show the robot trajectories of the proposed controller in case of white noise (assuming sensor noise) acting on the system output, the value of *M* is increased by 50% from the initial value. Figure 12 and Figure 13 present the circle and eight shape trajectories of the robot with I_v and I_w are increased by 50% from the initial value. The actual trajectories response of the robot still converges to the reference trajectory in a finite time with the error converges to zero. However, the control signals in Figure 11 oscillate more after increasing the structural parameters of the robot.

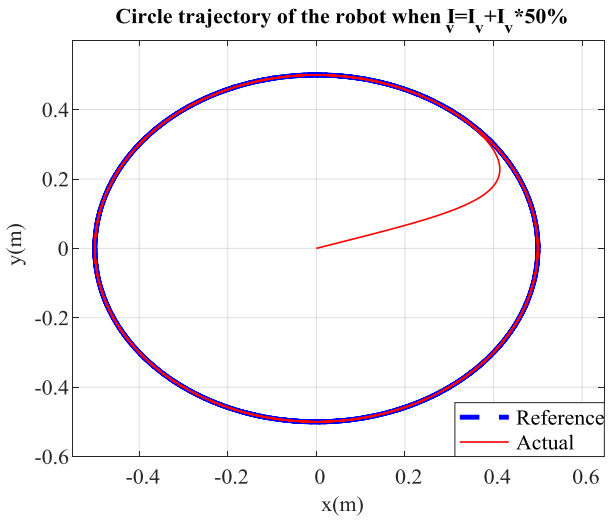


Figure 12. Robot circle trajectory when I_v is increased by 50%.

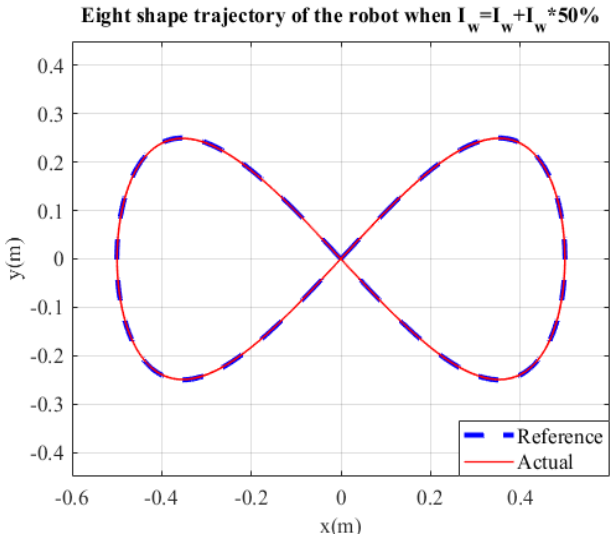


Figure 13. Robot eight shape trajectory when I_w is increased by 50%.

The results presented above show the effectiveness, suitability and robustness of the proposed control method in trajectory tracking control of the robot.

5. Conclusions

This paper presents the design of the PID sliding mode controller based on new Quasi-sliding mode (PID-SMC-NQ) and the radial basis function neural network (RBFNN) for Omni-directional mobile robot. The RBFNN is used to approximate the $\boldsymbol{A}_w$ matrix in the PID-SMC-NQ controller. The RBFNN is considered as an adaptive controller 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.. The weights of the network are trained online due to the feedback from output signals of the robot using the Gradient Descent algorithm. With this controller, the robot's actual trajectory follows the desired in a finite time with the rising time reaches 307.711 ms, 364.192 ms, the steady-state error is 0.0018 m and 0.00007 m, while the overshoot is 0.13% and 0.1% for the x_w and y_w, respectively, and reduces the chattering phenomena around the sliding surface. The quality criteria to evaluate the performance of the proposed controller are presented in Table 3 and Table 4 shows various error performance measures for each response. In the future, this research will use the Genetic Algorithm, or Particle Swarm Optimization, or Whale Optimization Algorithm to optimize the number of hidden layer nodes of the RBFNN and experiment with real models.

Conflict of interest

The authors declare there is no conflict of interest in this paper.

References

1. Ren C, Li C, Hu L, Li X, Ma S (2022) Adaptive model predictive control for an omnidirectional mobile robot with friction compensation and incremental input constraints. *T I Meas Control* 44: 835–847. https://doi.org/10.1177/01423312211021321
2. Hacene N, Mendil B (2019) Motion Analysis and Control of Three-Wheeled Omnidirectional Mobile Robot. *J Control Autom Electr Syst* 30: 194–213. https://doi.org/10.1007/s40313-019-00439-0
3. Palac ń J, Rubies E, Clotet E, Mart ńez D (2021) Evaluation of the Path-Tracking Accuracy of a Three-Wheeled Omnidirectional Mobile Robot Designed as a Personal Assistant. *Sensors* 21: 1–19. https://doi.org/10.3390/s21217216
4. Kawtharani MA, Fakhari V, Haghjoo MR (2020) Tracking Control of an Omni-Directional Mobile Robot. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications* (HORA*)*, 1–8. https://doi.org/10.1109/HORA49412.2020.9152835
5. Andreev AS, Peregudova OA (2020) On Global Trajectory Tracking Control for an Omnidirectional Mobile Robot with a Displaced Center of Mass. *Rus J Nonlin Dyn* 16: 115–131. https://doi.org/10.20537/nd200110
6. Mou H (2020) Research On the Formation Method of Omnidirectional Mobile Robot Based On Dynamic Sliding Mode Control. *Academic Journal of Manufacturing Engineering* 18: 148–154.
7. Nganga-Kouya D, Okou F, Lauhic Ndong Mezui JM (2021) Modeling and Nonlinear Adaptive Control for Omnidirectional Mobile Robot. *RJASET* 18: 59–69. https://doi.org/10.19026/rjaset.18.6064

8. Mehmood A, ul H. Shaikh I, Ali A (2021) Application of Deep Reinforcement Learning for Tracking Control of 3WD Omnidirectional Mobile Robot. *ITC* 50: 507–521. https://doi.org/10.5755/j01.itc.50.3.25979
9. Loucif F, Kechida S (2020) Optimization of sliding mode control with PID surface for robot manipulator by Evolutionary Algorithms. *Open Computer Science* 10: 396–407. https://doi.org/10.1515/comp-2020-0144
10. Liu J (2017) *Sliding Mode Control Using MATLAB*. Elsevier Science. https://doi.org/10.1016/B978-0-12-802575-8.00005-9
11. Li H, Huang S (2021) Research on the Prediction Method of Stock Price Based on RBF Neural Network Optimization Algorithm. *E3S Web Conf* 235: 1–5. https://doi.org/10.1051/e3sconf/202123503088
12. Wang H, Zhao Y, Pei J, Zeng D, Liu M (2019) Non-negative Radial Basis Function Neural Network in Polynomial Feature Space. *J Phys Conf Ser* 1168: 1–8. https://doi.org/10.1088/1742-6596/1168/6/062005
13. Lemita A, Boulahbel S, Kahla S (2020) Gradient Descent Optimization Control of an Activated Sludge Process based on Radial Basis Function Neural Network. *Eng Technol Appl Sci Res* 10: 6080–6086. https://doi.org/10.48084/etasr.3714
14. Kaya AI, İLkuçar M, ÇiFci A (2019) Use of Radial Basis Function Neural Network in Estimating Wood Composite Materials According to Mechanical and Physical Properties. *Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi* 12: 116–123. https://doi.org/10.18185/erzifbed.428763
15. Lemita A, Boulahbel S, Kahla S, Sedraoui M (2020) Auto-Control Technique Using Gradient Method Based on Radial Basis Function Neural Networks to Control of an Activated Sludge Process of Wastewater Treatment. *JESA* 53: 671–679. https://doi.org/10.18280/jesa.530510
16. Pham TT, Le MT, Nguyen CN (2021) Omnidirectional Mobile Robot Trajectory Tracking Control with Diversity of Inputs. *International Journal of Mechanical Engineering and Robotics Research* 10: 639–644. https://doi.org/10.18178/ijmerr.10.11.639-644
17. Mukherjee I, Routroy S (2012) Comparing the performance of neural networks developed by using Levenberg–Marquardt and Quasi-Newton with the gradient descent algorithm for modelling a multiple response grinding process. *Expert Syst Appl* 39: 2397–2407. https://doi.org/10.1016/j.eswa.2011.08.087

AIMS Press