International Journal of Applied Mathematics in Control Engineering 2 (2019) 118-126

Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.yxpublications.com/ijamce/index.html

Adaptive Tracking Control of Wheeled Mobile Robot Based on Neural Networks and Slip-compensation

Xingguo Song, Tingting Liu*

School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, 610031, China;

ARTICLE INFO

Article history:
Received 20 September 2019
Accepted 15 October 2019
Available online 25 October 2019

Keywords:
Wheeled mobile robot
Neural Networks
Slip ratio
Lyapunov theory

ABSTRACT

Wheeled Mobile Robots (WMRs) provide the advantage of stable locomotion on uneven terrain; hence, such mechanisms are used for locomotion by outdoor robots including those used for search and rescue. However, slippage often occurs when WMRs follows a slope in uneven terrain, and the slippage generates large accumulated positioning errors in the vehicle, compared with conventional wheeled mobile robots. An estimation of the wheel slip ratio is essential to improve the accuracy of locomotion control. In this paper, we propose an improved slip-compensation and adaptive NN control method to allow WMR to follow a slope curve, where stability is guaranteed by Lyapunov theory. All system states use neural network online weight tuning algorithms, which guarantee small tracking errors and no loss of stability in robot motion with bounded input signals. We demonstrate superior tracking results using the proposed control method in various Matlab simulations.

1. Introduction

In recent years, several researchers have investigated the problem of controlling nonholonomic systems. A major representative nonholonomic system, the wheeled mobile robot (WMR), provides the advantage of stable locomotion on uneven terrain during outdoor locomotion. Such robots are used for search and rescue, and they have attracted considerable attention in the last few years. Most authors concentrate on the nonholonomic problem that arises when side slip and tangential slip of the wheels are neglected (Kim et al., 2002).

Among these systems, many modern control theory techniques and stabilizing control algorithms have been proposed for the motion control of WMRs. A stable control scheme is proposed for an autonomous mobile robot under the assumption of perfect velocity tracking (Kanayama et al., 1990). A modified input-output linearization method was proposed to solve the problem of a decoupling matrix using a generalized inverse that provided a least-squares solution to the tracking control of two-wheeled mobile robots (Kim et al., 1999). A PID controller was proposed to solve the path tracking problem of a mobile robot using a simple linearized model of the mobile robot, which was composed of an integrator and a delay (Julio et al., 2001). An output-feedback controller was proposed that allowed a unicycle mobile robot to track a predefined path (Do et al., 2006). However, all of these control methods require a known and accurate mathematical model of nonholonomic mobile robot systems.

In practice, fully autonomous WMR control systems need to cope with dynamical robot uncertainties, unmodeled or unstructured disturbances, and nonlinear friction. Thus, it is difficult to obtain an accurate mathematical model for applying computed torque controllers or other model-based controllers. Many modern control techniques are designed to control the motion of WMRs; however, control performance is often degraded by modeling errors, information feedback errors, and external disturbances. The neural network control of mobile robots has been the subject of intense research in recent years (Wang et al., 2006, Li et al., 2009). This research has produced new methods for solving the main difficulties. Neural networks are most often used as function approximators. Wang investigated tracking control using an adaptive smart neural network for WMRs and they produced fine motion control based on partially unknown dynamics. A neural network-based model was presented that combined the backstepping technique with a torque controller (Fierro et al., 1997). The control methods stated above are designed under the constraint of pure roll and no slip.

However, slippage always exists between the wheels and ground (Ray et al., 2009) when a WMR moves on uneven terrain or undulating slopes. The slippage generates large numbers of accumulated positioning errors in the vehicle compared with conventional WMRs. Therefore, the control systems will be based on the effected by wheeled slippage as long as slippage exists when

* Corresponding author.
E-mail addresses: ttliu18@163.com (T. T. Liu)

a WMR is in motion; thus, the WMR motion control system will not be stable. In order to solve this problem, we propose an approach based on the estimation of each wheel's quantitative slippage, which improves the accuracy of the vehicle velocity. The analysis of wheel slippage is based on the "Theory of Ground Vehicles" (Wong, 1978). Ding showed that slip ratios of all wheels could be obtained by experimental study (Ding et al., 2010). Some methods were proposed for measuring the linear velocities of wheels and for detecting the current velocity of vehicle, which used encoders and a gyro-sensor, respectively (Ding et al., 2010, Endo et al., 2007). These methods proved useful for estimating the slip ratios of wheels. It is difficult to establish a precise mathematical model because of external disturbances and unmodeled or unknown system parameters. This means that we need an adaptive controller to solve these problems.

In this paper, the author proposes an effective adaptive robust motion tracking control method based on neural networks and slip-compensation for WMR systems. The control objective is to track a specified motion trajectory (e.g., deformable slope) in the proposed closed-loop system. This research identifies the model of a WMR system, assuming the appropriate slip ratio, and we characterize the unknown robot dynamics of the system. This paper developed the proposed control method using neural networks to model the system dynamics and nonlinearities. The radial basis function (RBF) neural network is well suited to uncertain or nonlinear functions because to its rapid online learning ability and nonlinear characteristics. A robust term can avoid the effects of external disturbances. Slip-compensation can decrease the error of tracking velocity. In this paper, the control approach includes radial basis function neural networks, proportional-differential (PD) control strategy, a robust term, and slip-compensation. The author use the control approach to overcome unknown system parameters and slippage disturbances in the WMR system, and to find a suitable velocity control input that stabilizes the closed-loop system. This controller guarantees perfect velocity tracking and the posture error converges to 0. The proposed control method facilitates precise motion tracking performance, which was demonstrated using Matlab simulation.

The remainder of this paper is organized as follows. The basics of nonholonomic systems slip ratios, and RBF neural networks are introduced in Section II. Section III discusses the robust adaptive NN control method based on slip-compensation, and Section IV covers the stability analysis. The Matlab simulation is presented in Section V. Finally, Section VI concludes the paper.

2. Preliminaries

2.1 Nonholonomic WMR Model

A mobile robot system in an n-dimensional configuration space C with the generalized coordinates $(q_1,\ldots,q_n)$ that is subject to m constraints can be described by (Li et al., 2010):

$$\boldsymbol{M}(q)\ddot{q}+\boldsymbol{C}(q,\dot{q})\dot{q}+\boldsymbol{F}(\dot{q})+\boldsymbol{G}(q)+\boldsymbol{\tau}_d=\boldsymbol{B}(q)\boldsymbol{\tau}-\boldsymbol{A}^{\mathrm{T}}(q)\lambda \quad (1)$$

where $\boldsymbol{M}(q)\in R^{n\times n}$ is a symmetric, positive definite inertia matrix, $\boldsymbol{C}(q,\dot{q})\in R^{n\times n}$ is the centripetal and Coriolis matrix, $\boldsymbol{F}(\dot{q})\in R^{n\times 1}$ denote friction coefficients, $\boldsymbol{G}(q)\in R^{n\times 1}$ denote gravitational torques (or forces), $\boldsymbol{\tau}_d$ denotes bounded external disturbances and unmodeled dynamics, $\boldsymbol{B}(q)\in R^{n\times r}$ is the input transformation matrix, $\boldsymbol{\tau}\in R^{r\times 1}$ are the input torques supplied by actuators, $\boldsymbol{A}(q)\in R^{m\times n}$ is the matrix associated with the constraints, and $\lambda\in R^{m\times 1}$ is the vector of constraint forces.

Property: $\dot{\boldsymbol{M}}-2\boldsymbol{C}$. is skew-symmetric: i.e.,

$$x^{\mathrm{T}}\left(\dot{\boldsymbol{M}}-2\boldsymbol{C}\right)x=0,\quad \forall x\neq 0.$$

The Euler-Lagrangian formulation is used to derive the dynamic equations for the mobile robot. Its potential energy U and its kinetic energy K is given by [15]:

$$U=\boldsymbol{M}(q)gh\ ,\ \ k_i=\frac{1}{2}m_iv_i^2+\frac{1}{2}w_i^{\mathrm{T}}I_iw_i\ ,$$

$$K=\sum_{i=1}^{n_i}k_i\ ,\ \ K=\frac{1}{2}\dot{q}^{\mathrm{T}}\boldsymbol{M}(q)\dot{q}$$

The dynamical equations of the mobile base in Fig. 1 can be expressed using a matrix form (1),

$$\boldsymbol{M}(q)=\begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix},$$

$$\boldsymbol{C}(q,\dot{q})=\begin{bmatrix} 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & 0 \end{bmatrix},\ \ \boldsymbol{G}(q)=\begin{bmatrix} mg\sin\varphi\cos\theta \\ mg\sin\varphi\sin\theta \\ 0 \end{bmatrix},$$

$$\boldsymbol{\tau}_d=\begin{bmatrix} \tau_{d_1} \\ \tau_{d_2} \end{bmatrix},\boldsymbol{B}(q)=\frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix},$$

$$\boldsymbol{A}^{\mathrm{T}}(q)=\begin{bmatrix} -\sin\theta \\ \cos\theta \\ -d \end{bmatrix},\ \ \lambda=-m(\dot{x}_c\cos\theta+\dot{y}_c\sin\theta)\dot{\theta}\ .$$

where m is the mass of the robot, I is the mass moment of inertia about the mass center $\boldsymbol{C}$, $2L$ is the distance between the driving wheels, and r is the radius of the wheel of the mobile robot. Here, θ is the main direction of the driving wheels of the vehicle, φ is the slope angle between forward orientation and horizontal surface (*X-O-Y*), d is the distance between the center $\boldsymbol{C}$ and the axis of the driving wheels. Previous studies generally assume that $\boldsymbol{G}(q)=\boldsymbol{0}$ ($\varphi=0$), i.e., the vehicle moves in a horizontal direction. Here, we consider $\boldsymbol{G}(q)$ can be not equal to 0, excluding this constraint condition.

We consider that all equality constraints of kinematics are independent of time and they can be expressed as follows:

$$\boldsymbol{A}(q)\dot{q}=0 \quad (2)$$

Let $\boldsymbol{S}(q)$ be a full rank matrix, $n-m$, formed by a set of smooth and linear independent vector fields spanning the null space of $\boldsymbol{A}(q)$:

$$\boldsymbol{S}^{\mathrm{T}}(q)\boldsymbol{A}^{\mathrm{T}}(q)=0 \quad (3)$$

According to (2) and (3), it is possible to find an auxiliary vector time function $\boldsymbol{v}(t)\in\Re^{n-m}$ such that for all t

$$\dot{q}=\boldsymbol{S}(q)\boldsymbol{v}(t) \quad (4)$$

The mobile robot shown in Fig. 1 is a typical example of a nonholonomic mechanical system. It has two parallel driving wheels mounted on the same axis. The motion and orientation are controlled by independent actuators, i.e., DC motors provide the necessary torque to the rear wheels. The stability of the platform is ensured by means of a front castor. The position of the robot in an inertial Cartesian frame {O, X, Y} is completely specified by the vector, where x_c, y_c are the coordinates of the center $\boldsymbol{C}$ of the mass and {C, X_c, Y_c} is the local coordinate with an origin of (x_c, y_c) with respect to the inertial basis.

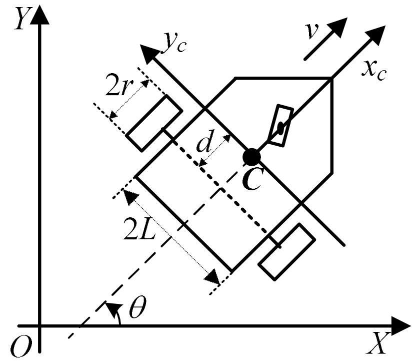

Fig.1. A nonholonomic mobile robot.

The nonholonomic constraint states that the robot can only move in a direction normal to the axis of the driving wheels. Previous studies always assumed that the system is subject to a "pure rolling without slipping" constraint (Lewis et al., 1993):

$$\dot{y}_c \cos\theta - \dot{x}_c \sin\theta - d\dot{\theta} = 0$$

It follows that $\boldsymbol{S}(q)$ is given by:

$$\boldsymbol{S}(q) = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \tag{5}$$

Under such a constraint condition, the vehicle is described by the following kinematic model:

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \boldsymbol{S}(q)\boldsymbol{v} \tag{6}$$

$$\ddot{q} = \dot{\boldsymbol{S}}(q)\boldsymbol{v} + \boldsymbol{S}(q)\dot{\boldsymbol{v}} \tag{7}$$

where $\boldsymbol{v} = \begin{bmatrix} v & \omega \end{bmatrix}^{\mathrm{T}}$, v and w are the displacement/linear and the angular velocities (which are also functions of time) respectively. In addition, $|v| \le v_{\max}$ and $|\omega| \le \omega_{\max}$, $v_{\max}$ and $\omega_{\max}$ are the maximum linear and angular velocities of the mobile robot.

However, when tangential slippage occurs between the wheels and the ground we adopt the following kinematics equations to reflect the effects of the slippage:

$$\dot{x} = \frac{v_1(1-s_1) + v_2(1-s_2)}{2}\cos\theta \tag{8}$$

$$\dot{y} = \frac{v_1(1-s_1) + v_2(1-s_2)}{2}\sin\theta \tag{9}$$

$$\dot{\theta} = \frac{v_1(1-s_1) - v_2(1-s_2)}{2L} \tag{10}$$

where v_1 and v_2 are the theoretical left- and right-wheel velocities, respectively, $2L$ is the tread, and s_1 and s_2 are the slip ratios defined as follows:

$$s_1 = \frac{v_1 - v_1'}{v_1}, \quad s_2 = \frac{v_2 - v_2'}{v_2}$$

where v_1' and v_2' are the factual/current velocity of the left and right wheels, respectively. If no slippage occurs, the current velocity is equal to the theoretical velocity ($v_1' = v_1$, $v_2' = v_2$), and the above equations (8), (9), (10) result in the conventional kinematics of WMRs.

2.2 Structural Properties of a Mobile Robot

According to the Euler-Lagrangian formulation, the dynamics of system (1) are now transformed into a more appropriate representation for control purposes. Using equations (4), (6), and (7), the complete equations of motion for the nonholonomic mobile platform are given by:

$$\boldsymbol{M}(q)(\dot{\boldsymbol{S}}(q)\boldsymbol{v} + \boldsymbol{S}(q)\dot{\boldsymbol{v}}) + \boldsymbol{C}(q,\dot{q})\boldsymbol{S}(q)\boldsymbol{v} + \boldsymbol{F}(\dot{q}) + \boldsymbol{G}(q) + \boldsymbol{\tau}_d$$
$$= \boldsymbol{B}(q)\boldsymbol{\tau} - \boldsymbol{A}^{\mathrm{T}}(q)\lambda$$

$$\boldsymbol{S}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{S}\dot{\boldsymbol{v}} + \boldsymbol{S}^{\mathrm{T}}(\boldsymbol{M}\dot{\boldsymbol{S}} + \boldsymbol{C}\boldsymbol{S})\boldsymbol{v} + \bar{\boldsymbol{F}} + \bar{\boldsymbol{G}} + \bar{\boldsymbol{\tau}}_d = \boldsymbol{S}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\tau} \tag{11}$$

Using appropriate definitions, we can rewrite as the following equation:

$$\bar{\boldsymbol{M}}(q)\dot{\boldsymbol{v}} + \bar{\boldsymbol{C}}(q,\dot{q})\boldsymbol{v} + \bar{\boldsymbol{F}}(\boldsymbol{v}) + \bar{\boldsymbol{G}}(\boldsymbol{v}) + \bar{\boldsymbol{\tau}}_d = \bar{\boldsymbol{\tau}} \tag{12}$$

$$\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{B}}\boldsymbol{\tau}$$

where $\bar{\boldsymbol{M}}(q) \in \Re^{r\times r}$ is a symmetric positive definite inertia matrix, $\bar{\boldsymbol{C}}(q,\dot{q}) \in \Re^{r\times r}$ is the centripetal and Coriolis matrix, $\bar{\boldsymbol{F}}(\boldsymbol{v}) \in \Re^{r\times 1}$ is the surface friction, $\bar{\boldsymbol{G}}(\boldsymbol{v}) \in R^{r\times 1}$ denote gravitational torques (or forces), $\bar{\boldsymbol{\tau}}_d$ denotes bounded unknown disturbances including unstructured, unmodeled dynamics, and $\|\bar{\boldsymbol{\tau}}_d\| \le b_d$, and $\bar{\tau} \in \Re^{r\times 1}$ is the input vector. If $r = n - m$, it follows that $\bar{\boldsymbol{B}}$ is a constant, nonsingular matrix that depends on the distance between the driving wheels, L, and the radius of the wheel, r. Equation (6) describes the nonholonomic mobile robot in local coordinates attached to its center of mass, while $\boldsymbol{S}^{\mathrm{T}}(q)$ is a Jacobian matrix that transforms velocities $\boldsymbol{v}$ in the local coordinates to the constrained velocities $\dot{q}$ in Cartesian coordinates.

Skew-symmetry: The matrix $\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{C}}$ is skew-symmetric.

Under pure rolling without slipping, the linear and angular velocities of the WMR system can be denoted by:

$$\boldsymbol{v} = \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{bmatrix} \dfrac{v_1 + v_2}{2} \\ \dfrac{v_1 - v_2}{2L} \end{bmatrix}$$

If slippage occurs between the wheels and the ground during movements, the factual velocity is not equal to $\boldsymbol{v}$, but instead

$$\boldsymbol{v}_f = \begin{pmatrix} v_f \\ \omega_f \end{pmatrix} = \begin{bmatrix} \dfrac{v_1(1-s_1) + v_2(1-s_2)}{2} \\ \dfrac{v_1(1-s_1) - v_2(1-s_2)}{2L} \end{bmatrix} \tag{13}$$

where v_1, v_2 are the linear velocities of the right and left wheels, which can be measured by encoders, while s_1 and s_2 are the slip ratios of the right and left wheels, respectively.

The factual velocity $\boldsymbol{v}_f$ can be detected by motion capture using a gyro-sensor. Thus, according to (13) we can obtain the slip ratios of the right and left wheels as follows:

$$s_1 = 1 - \frac{v_f + \omega_f L}{v_1}, \quad s_2 = 1 - \frac{v_f - \omega_f L}{v_2} \tag{14}$$

2.3 RBF Neural Networks

Many different models of neural networks have been established (Karayiannis et al., 1997, Jung et al., 2001, Gao et al., 2019). They are mostly designed for specific objectives. Of these models, the RBF neural network is well suited to modeling uncertain or nonlinear functions. A typical RBF neural network is shown in Fig.2. It is a two-layer network comprised of a hidden layer and an output layer. The hidden layer consists of an array of functions, i.e., RBFs, while the output layer is merely a linear combination of the hidden layer functions. Using this simple structure, the RBF neural network facilitates a more effective weight updating procedure compared with other, more complex multilayer networks.

In this study, the RBF neural network was used for function approximation. We assume a smooth function $g(\boldsymbol{\theta}):\Re^p \to \Re^m$ expressed in terms of the RBF neural network as follows:

$$g(\boldsymbol{\theta}) = \boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta}) \tag{15}$$

$$\boldsymbol{W}^{\mathrm{T}} = \begin{bmatrix} w_{10} & w_{11} & w_{12} & \cdots & w_{1L} \\ w_{20} & w_{21} & w_{22} & \cdots & w_{2L} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{m0} & w_{m1} & w_{m2} & \cdots & w_{mL} \end{bmatrix}$$

$$\phi(\boldsymbol{\theta}) = \begin{bmatrix} 1 & \phi_1(\boldsymbol{\theta}) & \phi_2(\boldsymbol{\theta}) & \cdots & \phi_L(\boldsymbol{\theta}) \end{bmatrix}^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{\theta} \in \Re^p$ is the input vector, $\boldsymbol{W} \in \Re^{(L+1)\times m}$ contains the ideal thresholds $w_{10}\cdots w_{m0}$, and weights $w_{11}\cdots w_{mL}$, of the neural network, $\phi(\boldsymbol{\theta}) \in \Re^{(L+1)}$ is the activation vector comprising the RBFs, and $\varepsilon(\boldsymbol{\theta}) \in \Re^m$ is a vector of the neural network function approximation errors.

Using the neural network shown in Fig.2 with a sufficiently large number (L) of RBFs in the hidden layer to approximate the smooth function $g(\boldsymbol{\theta})$ described by (11), there exist positive numbers b_w and b_ε such that:

$$\|\boldsymbol{W}\|_F \le b_w \quad \text{and} \quad \|\boldsymbol{\varepsilon}(\boldsymbol{\theta})\| \le b_\varepsilon \quad \forall \boldsymbol{\theta} \tag{17}$$

where the symbols $\|\bullet\|_F$ and $\|\bullet\|$ denote the Frobenius matrix norm and Euclidean vector norm, respectively (Hom et al., 1985). For control purposes, we only need to know that these *ideal* approximating NN weights exist for a specified value of b_ε. Thus, an estimate of $g(\boldsymbol{\theta})$ can be given by:

$$\hat{g}(\boldsymbol{\theta}) = \hat{\boldsymbol{w}}^{\mathrm{T}}\phi(\boldsymbol{\theta})$$

where $\hat{\boldsymbol{w}}$ is an estimate of $\boldsymbol{w}$.

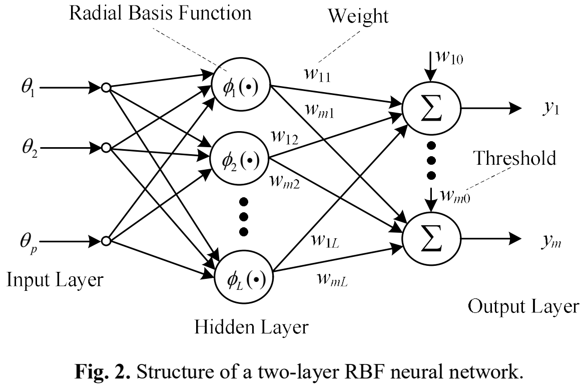

Fig. 2. Structure of a two-layer RBF neural network.

A suitable RBF may be selected from a large class of functions for the activation vector (12). A Gaussian function is typically selected and the RBF $\phi_i(\boldsymbol{\theta})$ for $i = 1,\ldots,L$ in (12) is given by:

$$\phi_i(\boldsymbol{\theta}) = \exp\left[-\frac{1}{2\sigma_i^2}\|\boldsymbol{\theta} - \mu_i\|^2\right]$$

where μ_i is the mean or center of the function, and σ_i denotes its width.

A common weight-tuning algorithm is the gradient algorithm based on the *backpropagated* error (Haykin, 2009). Here, the NN is trained to match specified exemplar pairs (x, y), where x is the ideal NN input that yields the desired NN output y. The online tuning backpropagation algorithm for the two layers is designated as:

$$\dot{\hat{w}}_{ij} = \eta_1\left(y_i - g_i(\boldsymbol{\theta})\right)\phi_j(x)$$

$$\dot{\mu}_j = \eta_2\phi_j(x)\frac{x-\mu_j}{\sigma_j^2}\sum_{i=1}^{l}\hat{w}_{ij}\left(y_i - g_i(\boldsymbol{\theta})\right)$$

$$\dot{\sigma}_j = \eta_3\phi_j(x)\frac{\|x-\mu_j\|^2}{\sigma_j^3}\sum_{i=1}^{l}\hat{w}_{ij}\left(y_i - g_i(\boldsymbol{\theta})\right)$$

where $\eta_1,\ \eta_2,\ \eta_3 > 0$ are the NN learning laws of w_i, μ_i, and σ_i, respectively.

3. Robust Adaptive NN Motion Controller Design

3.1Motion tracking problem

In this control system, we use two postures: the *reference posture* $q_r = (x_r, y_r, \theta_r)^{\mathrm{T}}$ and the *current posture* $q = (x, y, \theta)^{\mathrm{T}}$. A reference posture is a goal posture of the vehicle and a current posture is its "real" posture at any given moment. The tracking position error vector is expressed on the basis of a frame linked to the mobile robot [2] as:

$$\boldsymbol{e}_m = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = \boldsymbol{T}_e\boldsymbol{e} \tag{18}$$

We can prove that the time derivative of the above position error is given by:

$$\dot{\boldsymbol{e}}_m = \begin{bmatrix} \omega y_e - v + v_r\cos\theta_e \\ -\omega x_e + v_r\sin\theta_e \\ \omega_r - \omega \end{bmatrix} \tag{19}$$

1) When the slip is neglected in the control design, an auxiliary velocity control law input that achieves tracking for (3) is given by

$$\boldsymbol{v}_c = \begin{bmatrix} v_r\cos\theta_e + k_1x_e \\ \omega_r + k_2v_ry_e + k_3v_r\sin\theta_e \end{bmatrix} \tag{20}$$

where $k_1,\ k_2,\ k_3 > 0$ are the feedback gains of x_e, y_e, and θ_e, respectively. If we consider only the kinematic model of the mobile robot (6) with its velocity input (20) and we assume perfect velocity tracking, then the kinematic model is *uniformly asymptotically stable* with respect to a reference trajectory (i.e., $e_m \to 0$ as $t \to \infty$).

The theoretical velocity generates a tracking error in the inner closed-loop, which is defined as follows:

$$\boldsymbol{e}_c = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \boldsymbol{v}_c - \boldsymbol{v} \tag{21}$$

2) However, when the slippage occurs between the wheels and the ground, the actual velocity of the vehicle is not equal to the desired velocity due to the wheeled slippage even if the theoretical velocity

satisfies $v=v_c$, $\omega=\omega_c$, such that the real position is expected but not equal to the desired position. Therefore, if we do not consider the slippage between the wheels and the ground, the designed controller will be unstable and it cannot track the reference trajectory. To solve this problem, we propose adding a slip-compensation method in the design of the controller. If the slip ratios of both wheels are detected quantitatively, this controller can be implemented to consider the slippage.

Here, we need to compensate for the loss of velocity caused by wheel slippage while maintaining the desired velocity of the WMR when we design the controller. Therefore, the velocity control input should be compensated based on the estimation of slip ratios using Eq. (22)

$$\boldsymbol{v}_{c1}=\begin{pmatrix} v_{c1} \\ \omega_{c1} \end{pmatrix}=\begin{bmatrix} (\frac{\upsilon_1}{1-\hat{s}_1}+\frac{\upsilon_2}{1-\hat{s}_2})/2 \\ (\frac{\upsilon_1}{1-\hat{s}_1}-\frac{\upsilon_2}{1-\hat{s}_2})/2L \end{bmatrix} =\begin{bmatrix} (\frac{v_c+\omega_c L}{1-\hat{s}_1}+\frac{v_c-\omega_c L}{1-\hat{s}_2})/2 \\ (\frac{v_c+\omega_c L}{1-\hat{s}_1}-\frac{v_c-\omega_c L}{1-\hat{s}_2})/2L \end{bmatrix} \tag{22}$$

where $\upsilon_1=v_c+\omega_c L$ and $\upsilon_2=v_c-\omega_c L$, respectively, are the linear velocities from the control input of the right and left wheel before slip-compensation, while $\hat{s}_1$ and $\hat{s}_2$ are the estimated slip ratios of the right and left wheels, which can be obtained by equality (14).

After slip-compensation, the velocity tracking error is defined as follows:

$$\boldsymbol{e}_{c1}=\begin{bmatrix} e_3 \\ e_4 \end{bmatrix}=\boldsymbol{v}_{c1}-\boldsymbol{v} \tag{23}$$

Differentiating (23), using (12), the mobile robot dynamics may be written in terms of the velocity tracking error as:

$$\bar{\boldsymbol{M}}(q)\dot{e}_{c1}=-\bar{\boldsymbol{C}}(q,\dot{q})\boldsymbol{e}_{c1}-\bar{\boldsymbol{\tau}}+g(\boldsymbol{x})+\bar{\boldsymbol{\tau}}_d \tag{24}$$

where the important nonlinear mobile robot function is defined as:

$$g(\boldsymbol{x})=\bar{\boldsymbol{M}}\dot{\boldsymbol{v}}_{c1}+\bar{\boldsymbol{C}}(q,\dot{q})\boldsymbol{v}_{c1}+\bar{\boldsymbol{F}}(\boldsymbol{v})+\bar{\boldsymbol{G}}(\boldsymbol{v}) \tag{25}$$

Here, the vector x can be measured by $\boldsymbol{x}\equiv\begin{bmatrix} \boldsymbol{v}_c^{\mathrm{T}} & \dot{\boldsymbol{v}}_c^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$.

Function $g(\boldsymbol{x})$ contains all the mobile robot parameters, such as mass, moments of inertia, and friction coefficients. The nonlinear robot function $g(\boldsymbol{x})$ is often imperfectly known in applications and it is difficult to determine.

3.2 RBF Neural Network Control Scheme

Many approaches exist for selecting a velocity control $\boldsymbol{v}$, to be used in the steering system of a mobile robot (6). In this section, we seek to convert the prescribed control $\boldsymbol{v}$ into a torque control $\boldsymbol{\tau}$ for the actual physical WMR. Therefore, we designed an NN control algorithm to produce the desired behavior, which is moved by the specific choice of velocity, $\boldsymbol{v}$.

In previous studies, the nonholonomic tracking problem is simplified by neglecting the vehicle dynamics (12) and considering only the steering system (6). Thus, a steering system input $\boldsymbol{v}_c$ is determined such that (6) tracks the reference cart trajectory. Here, it is assumed that there is "perfect velocity tracking" so that $\boldsymbol{v}=\boldsymbol{v}_{c1}$, and then (11) is used to compute $\boldsymbol{\tau}$. This approach has three problems: first, the perfect velocity tracking assumption does not hold in practice; second, the disturbance $\boldsymbol{\tau}_d$ is ignored; third, the approach requires complete knowledge of the dynamics. However, "perfect velocity tracking" can be performed effectively if we only consider the vehicle's kinematic model. Therefore, we use the *NN integrator backstepping method* to deal with unmodeled bounded disturbances and unstructured, unmodeled dynamics in the vehicle, such that $\boldsymbol{v}=\boldsymbol{v}_{c1}$.

In applications, the nonlinear robot function $g(\boldsymbol{x})$ is at least partially unknown. A suitable control input for velocity following is given by a neural network computed-torque controller. The unknown system function $g(\boldsymbol{x})$ is approximated using the RBF neural network described by (11). A major advantage is that this can always be accomplished, due to the RBF NN approximation property. To consider the function $g(\boldsymbol{x})$ given by (25), the vector in (16) is defined as:

$$\boldsymbol{\theta}\equiv\begin{bmatrix} \boldsymbol{v}_{c1}^{\mathrm{T}} & \dot{\boldsymbol{v}}_{c1}^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{26}$$

The neural network (15) for (25) is redefined as $g(\boldsymbol{\theta}):\Re^6\rightarrow\Re^2$, which is rewritten as:

$$g(\boldsymbol{\theta})=\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{\theta})+\varepsilon(\boldsymbol{\theta}) \tag{27}$$

where $\boldsymbol{w}\in\Re^{(L+1)}$ is the vector of the ideal threshold and their weights. The bounds described by (17) are modified for $\boldsymbol{w}$ and $\varepsilon(\boldsymbol{\theta})$ and expressed as:

$$\|\boldsymbol{w}\|\le b_w \text{ and } |\varepsilon(\boldsymbol{\theta})|\le b_\varepsilon \quad \forall\boldsymbol{\theta}. \tag{28}$$

The configuration of the proposed robust NNC system, which combines a NN controller, PD controller, and robust term, is summarized in Fig. 3. The NN controller is connected in parallel with the PD controller and robust term to generate a compensated control signal. They form the inner closed-loop system that controls the velocity error, $e_{c1}\rightarrow 0$, while the outer closed-loop system controls the position/trajectory error, $e\rightarrow 0$ based on slip-compensation.

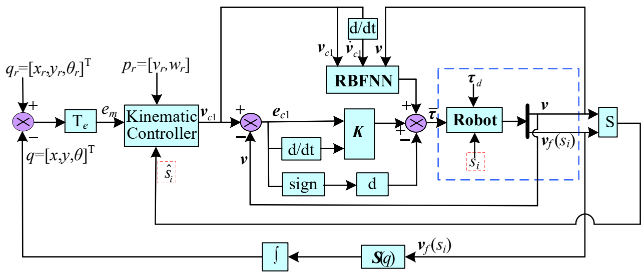

Fig. 3. Structure of the proposed robust adaptive neural network motion control methodology.

The control law is given by:

$$\bar{\boldsymbol{\tau}}=\hat{g}+\boldsymbol{KE}+d\,\mathrm{sgn}(\boldsymbol{e}_{c1}) \tag{29}$$

where $\hat{g}$ is an estimate of g and the output torque of NNC, $\boldsymbol{KE}$ is the torque produced by the PD controller, and $d\,\mathrm{sgn}(\boldsymbol{e}_{c1})$ is the robust term. An estimate of $g(\boldsymbol{\theta})$ can be given by:

$$\hat{g}(\boldsymbol{\theta})=\hat{\boldsymbol{w}}^{\mathrm{T}}\hat{\phi}(\boldsymbol{\theta}) \tag{30}$$

where $\hat{\boldsymbol{w}}\in\Re^{(L+1)}$ is the vector of the estimated threshold and weights. There is no simple, standard measurement to judge which choice is the best, hence, our assumption of $\hat{\phi}=\phi$ is always feasible. The velocity error vector is defined as:

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{v}_{c1} - \boldsymbol{v} \\ \dot{\boldsymbol{v}}_{c1} - \dot{\boldsymbol{v}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_{c1} \\ \dot{\boldsymbol{e}}_{c1} \end{bmatrix} \tag{31}$$

where $\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}_P & \boldsymbol{K}_D \end{bmatrix}$ and $\boldsymbol{K}_P = diag(k_4, k_5)$ as well as $\boldsymbol{K}_D = \begin{bmatrix} k_6 & k_7 \end{bmatrix}$ are positive real numbers. ***KE*** is given by:

$$\boldsymbol{KE} = \boldsymbol{K}_P \boldsymbol{e}_{c1} + \boldsymbol{K}_D \dot{\boldsymbol{e}}_{c1} \tag{32}$$

Let the NN weights be further adjusted to minimize the velocity tracking error. The adaptive law of $\hat{\boldsymbol{w}}$ is designated as:

$$\dot{\hat{w}}_i = \phi \boldsymbol{e}_{c1}^{\mathrm{T}} - \kappa \left\| \boldsymbol{e}_{c1} \right\| \hat{w}_i , \quad i = 1, 2, \ldots, n \tag{33}$$

where $\hat{\boldsymbol{w}} = \begin{bmatrix} \hat{w}_1 & \hat{w}_2 & \ldots & \hat{w}_n \end{bmatrix}$, *n* is the NN output, and κ is a positive constant. The center of function μ_i and its width σ_i use the gradient algorithm based on the *backpropagated* error.

The simultaneous updates of all three sets of parameters may be suitable for non-stationary environments or online settings.

Using the estimated function $\hat{g}(\boldsymbol{\theta})$ given by (30) and (32), the control law (29) becomes:

$$\boldsymbol{\bar{\tau}} = \hat{\boldsymbol{w}}^{\mathrm{T}} \phi(\boldsymbol{\theta}) + \boldsymbol{K}_P \boldsymbol{e}_{c1} + \boldsymbol{K}_D \dot{\boldsymbol{e}}_{c1} + d \,\mathrm{sgn}(\boldsymbol{e}_{c1}) \tag{34}$$

and the parameter *d* is defined as:

$$d \geq b_\varepsilon + b_d + \frac{1}{4} \kappa b_w^2 + \epsilon$$

which is related to the bounds described by (28), the parameter κ in (33), and a strictly positive constant ϵ .

4. Stability Analysis

In this section, we perform system stability analysis of the closed-loop behavior in the proposed control methodology. Thus, we derive and analyze the closed-loop dynamics. Substituting the control input (34) into the mobile robot dynamics system described by (24) yields:

$$(\bar{\boldsymbol{M}} + \boldsymbol{K}_D) \dot{\boldsymbol{e}}_{c1} = -(\boldsymbol{K}_P + \bar{\boldsymbol{C}}) \boldsymbol{e}_{c1} + \tilde{g} + \boldsymbol{\bar{\tau}}_d - d \,\mathrm{sgn}(\boldsymbol{e}_{c1}) \tag{35}$$

where $\tilde{g} = g - \hat{g}$ is the function estimation error. This estimation error is expressed according to (21) and (22) as:

$$\tilde{g}(\boldsymbol{\theta}) = \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta})$$

where $\tilde{\boldsymbol{w}}$ is the vector of the threshold and weight estimation errors, defined as:

$$\tilde{\boldsymbol{w}} = \boldsymbol{w} - \hat{\boldsymbol{w}}$$

Therefore, (29) can be written as follows:

$$\begin{aligned}(\bar{\boldsymbol{M}} + \boldsymbol{K}_D) \dot{\boldsymbol{e}}_{c1} &= -(\boldsymbol{K}_P + \bar{\boldsymbol{C}}) \boldsymbol{e}_{c1} + \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi(\boldsymbol{\theta}) \\ &\quad + \varepsilon(\boldsymbol{\theta}) + \boldsymbol{\bar{\tau}}_d - d \,\mathrm{sgn}(\boldsymbol{e}_{c1})\end{aligned}$$

Consider the following Lyapunov function candidate:

$$V = V_1(\boldsymbol{e}_{c1}, t) + V_2(\boldsymbol{e}_m, t)$$

where

$$V_1(\boldsymbol{e}_{c1}, t) = \frac{1}{2} \boldsymbol{e}_{c1}^{\mathrm{T}} \bar{\boldsymbol{M}} \boldsymbol{e}_{c1} + \frac{1}{2} \boldsymbol{e}_{c1}^{\mathrm{T}} \boldsymbol{K}_D \boldsymbol{e}_{c1} + \frac{1}{2} \left[(\boldsymbol{w} - \hat{\boldsymbol{w}})^{\mathrm{T}} (\boldsymbol{w} - \hat{\boldsymbol{w}}) \right]$$

Differentiation yields:

$$\dot{V}_1(\boldsymbol{e}_{c1}, t) = \boldsymbol{e}_{c1}^{\mathrm{T}} \bar{\boldsymbol{M}} \dot{\boldsymbol{e}}_{c1} + \frac{1}{2} \boldsymbol{e}_{c1}^{\mathrm{T}} \dot{\bar{\boldsymbol{M}}} \boldsymbol{e}_{c1} + \boldsymbol{e}_{c1}^{\mathrm{T}} \boldsymbol{K}_D \dot{\boldsymbol{e}}_{c1} + \mathrm{tr}\left[(\boldsymbol{w} - \hat{\boldsymbol{w}})^{\boldsymbol{T}} \dot{\hat{\boldsymbol{w}}} \right]$$

$$\begin{aligned}&= \boldsymbol{e}_{c1}^{\mathrm{T}} \left(-\bar{\boldsymbol{C}} \boldsymbol{e}_{c1} + \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi - \boldsymbol{K}_D \dot{\boldsymbol{e}}_{c1} - \boldsymbol{K}_P \boldsymbol{e}_{c1} - d \,\mathrm{sgn}(\boldsymbol{e}_{c1}) + \varepsilon + \boldsymbol{\bar{\tau}}_d \right) \\ &\quad + \frac{1}{2} \boldsymbol{e}_{c1}^{\mathrm{T}} \dot{\bar{\boldsymbol{M}}} \boldsymbol{e}_{c1} + \boldsymbol{e}_{c1}^{\mathrm{T}} K_D \dot{\boldsymbol{e}}_{c1} - \mathrm{tr}\left[\tilde{\boldsymbol{w}}^{\mathrm{T}} \left(\phi \boldsymbol{e}_{c1}^{\mathrm{T}} - \kappa \left\| \boldsymbol{e}_{c1} \right\| \hat{\boldsymbol{w}} \right) \right]\end{aligned}$$

$$\begin{aligned}&= \frac{1}{2} \boldsymbol{e}_{c1}^{\mathrm{T}} \left(\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{C}} \right) \boldsymbol{e}_{c1} + \boldsymbol{e}_{c1}^{\mathrm{T}} \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi - \boldsymbol{e}_{c1}^{\mathrm{T}} K_P \boldsymbol{e}_{c1} - d \left\| \boldsymbol{e}_{c1} \right\| + \boldsymbol{e}_{c1}^{\mathrm{T}} \left(\varepsilon + \boldsymbol{\bar{\tau}}_d \right) \\ &\quad - tr\left(\tilde{\boldsymbol{w}}^{\mathrm{T}} \phi \boldsymbol{e}_{c1}^{\mathrm{T}} - \kappa \tilde{\boldsymbol{w}}^{\mathrm{T}} \left\| \boldsymbol{e}_{c1} \right\| \left\| \hat{\boldsymbol{w}} \right\| \right)\end{aligned}$$

$$\begin{aligned}&= \boldsymbol{e}_{c1}^{\mathrm{T}} \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi - \boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - d \left\| \boldsymbol{e}_{c1} \right\| + \boldsymbol{e}_{c1}^{\mathrm{T}} \left(\varepsilon + \boldsymbol{\bar{\tau}}_{\boldsymbol{d}} \right) - tr\left(\boldsymbol{e}_{c1}^{\mathrm{T}} \tilde{\boldsymbol{w}}^{\mathrm{T}} \phi \right) \\ &\quad + tr\left[\kappa \tilde{\boldsymbol{w}} \left\| \boldsymbol{e}_{c1} \right\| (\boldsymbol{w} - \tilde{\boldsymbol{w}}) \right]\end{aligned}$$

$$\begin{aligned}&\leq -\boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - d \left\| \boldsymbol{e}_{c1} \right\| + \left\| \boldsymbol{e}_{c1} \right\| \left(\varepsilon + \boldsymbol{\bar{\tau}}_{\boldsymbol{d}} \right) - \kappa \left\| \boldsymbol{e}_{c1} \right\| \left\| \tilde{\boldsymbol{w}} \right\|^2 \\ &\quad + \kappa \left\| \boldsymbol{e}_{c1} \right\| \left\| \tilde{\boldsymbol{w}} \right\| b_w\end{aligned}$$

$$\begin{aligned}&= -\boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - d \left\| \boldsymbol{e}_{c1} \right\| + \left\| \boldsymbol{e}_{c1} \right\| \left(b_\varepsilon + b_d \right) - \kappa \left\| \boldsymbol{e}_{c1} \right\| \left(\left\| \tilde{\boldsymbol{w}} \right\|^2 - \left\| \tilde{\boldsymbol{w}} \right\| b_w \right) \\ &\quad - \frac{1}{4} \kappa b_w^2 \left\| \boldsymbol{e}_{c1} \right\| + \frac{1}{4} \kappa b_w^2 \left\| \boldsymbol{e}_{c1} \right\|\end{aligned}$$

$$\begin{aligned}&= -\boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - d \left\| \boldsymbol{e}_{c1} \right\| + \left\| \boldsymbol{e}_{c1} \right\| \left(b_\varepsilon + b_d \right) - \kappa \left\| \boldsymbol{e}_{c1} \right\| (\left\| \tilde{\boldsymbol{w}} \right\|^2 - \left\| \tilde{\boldsymbol{w}} \right\| b_w \\ &\quad + \frac{1}{4} b_w^2) + \frac{1}{4} \kappa b_w^2 \left\| \boldsymbol{e}_{c1} \right\|\end{aligned}$$

$$= -\boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - \left\| \boldsymbol{e}_{c1} \right\| \left\{ d - b_\varepsilon - b_d + \kappa \left[\left(\left\| \tilde{\boldsymbol{w}} \right\| - \frac{1}{2} b_w \right)^2 - \frac{1}{4} b_w^2 \right] \right\}$$

$$\leq -\boldsymbol{K}_P \left\| \boldsymbol{e}_{c1} \right\|^2 - \left\| \boldsymbol{e}_{c1} \right\| \left[\epsilon + \kappa \left(\left\| \tilde{\boldsymbol{w}} \right\| - \frac{1}{2} b_w \right)^2 \right] \tag{36}$$

Thus, $V_1 \geq 0$, $\dot{V}_1 \leq 0$ are guaranteed to be negative, and this shows that $V_1 \to 0$ and implies $\boldsymbol{e}_{c1} \to 0$ while $\dot{\boldsymbol{e}}_{c1} \to 0$ as $t \to \infty$. Furthermore, (36) shows $\dot{V}_1 = 0$ if and only if $\boldsymbol{e}_{c1} = 0$. Therefore, $\boldsymbol{v} \to \boldsymbol{v}_{c1}$, which equals $\boldsymbol{v}_f \to \boldsymbol{v}_c$ as $t \to \infty$. We continue by choosing a Lyapunov functional candidate:

$$V_2(\boldsymbol{e}_m, t) = \frac{1}{2}(x_e^2 + y_e^2) + (1 - \cos\theta_e) / k_2$$

Differentiation yields:

$$\dot{V}_2(\boldsymbol{e}_m, t) = -k_1 x_e^2 - k_3 \sin^2\theta_e / k_2 \leq 0 \tag{37}$$

According to (36) and (37), we can obtain $\dot{V} = \dot{V}_1 + \dot{V}_2 \leq 0$. Therefore, global stability is guaranteed by the standard Lyapunov theory. The system stability and velocity tracking convergence are guaranteed by the control law (34) driving the system (11), which closely tracks the desired motion trajectories.

If there are no slip ratios between the wheels and the ground, we only need to change $\boldsymbol{v}_{c1}$ into $\boldsymbol{v}_c$, and $\boldsymbol{e}_{c1}$ change into $\boldsymbol{e}_c$. This case indicates WMR under pure rolling without slipping; hence, we do not need to compensate for slippage, and the stability can be proved using the same method stated above.

Remark: In practice, the velocity and tracking errors are not exactly equal to zero when using control law (34). The best we can guarantee is that the error converges in the neighborhood of the origin. The discontinuous function “sign” will give rise to control chattering due to imperfect switching in the computer control. This is undesirable, because the unmodeled high-frequency dynamics might become excited. To avoid this effect, we apply boundary layer technique (Wu et al., 2018) to smooth the control signal. In a small neighborhood of the velocity error $(\boldsymbol{e}_{c1}=0)$, the discontinuous function "sign" is replaced by a boundary saturation function $\mathrm{sat}(e_i/\delta),\ i=3,4$, defined as:

$$\mathrm{sat}(e_i/\delta)=\begin{cases}-1 & : e_i<-\delta \\ e_i/\delta & : -\delta\le e_i\le\delta \\ +1 & : e_i>\delta\end{cases} \tag{38}$$

where δ is the specified boundary layer thickness. Thus, based on dynamics control, the robust neural network motion tracking control law (34) becomes:

$$\overline{\boldsymbol{\tau}}=\hat{\boldsymbol{w}}^{\mathrm{T}}\phi(\theta)+\boldsymbol{K}_P\boldsymbol{e}_{c1}+\boldsymbol{K}_D\dot{\boldsymbol{e}}_{c1}+d\,\mathrm{sat}(\boldsymbol{e}_{c1}/\delta) \tag{39}$$

5. Simulation Results

A wheel's slip ratio is an important state variable. When slippage occurs between the wheels and the ground, most of the wheel velocities are influenced by the slip ratio, necessitating their analysis using a certain slip ratio for movement control. According to the definition of the slip ratio, the range of slip ratios is from -1 to 1 [11]. According to [11] and [12], we know that the change in slip ratios is similar to a sine curve and the slip ratio will increase with increasing slope. Therefore, we assume that the reference trajectory is a space circle, where $x_r=3\cos(t/3)$, $y_r=3\sin(t/3)$, $z_r=-0.4\cos(2t/3)$, and $t\in[0,\ 20]$ and where the initial position is $Q=(3,0,0)$. The slip ratio can be considered as the time variable function "0.4sin(2*t*/3)", which increases or decreases with the slope angle of the reference trajectory.

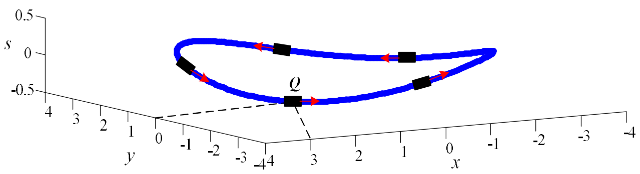

Fig. 4. Reference circle trajectory

We now demonstrate the adaptive robust NN control shown in Fig. 3. Here, we compare its performance without slippage and with slippage between the wheels and the ground, before we compensate using the slip ratios for this NN controller, which is indicated by the left dashed circle in Fig.4. Three control performances were implemented and tested using Matlab Simulink models: 5.1. Robust NN controller without slippage; 5.2. Robust NN controller with slippage; 5.3. Robust NN controller with slip-compensation. We adopt vehicle parameters (Fig. 1) of $m=10\mathrm{kg}$, $I=5\mathrm{kgm}^2$, $L=0.2\mathrm{m}$, $d=0\mathrm{m}$, $r=0.05\mathrm{m}$, under the time varying external disturbance $\overline{\tau}_d=(\sin t+\cos t,\ 1)^{\mathrm{T}}$, and $v_r=1.0\mathrm{m/s}$, $\omega_r=1/3\,\mathrm{rad/s}$. The objective is to track the trajectory such that the errors in the position and velocity tend to zero. The controller gains were selected so that the closed-loop system exhibits a critical damping behavior: $k_1=1,\ k_2=9,\ k_3=6,\ \boldsymbol{K}=diag\{20,10,10,5\}$. In the neural network, we selected a radial basis function with $N_h=6$ hidden-layer neurons and $\kappa=0.35$.

5.1 Robust NN controller without slippage

The response of this controller is described under conditions of pure roll and without slip, as shown in Fig.5. The results show that this robust NN controller was effective, even when bounded unmodeled disturbances and non-symmetric friction were included. The performance of the system was clearly improved compared with previous methods. Moreover, the NN controller requires no prior information about the dynamics of the vehicle. Like the classical PID torque controller, the NN controller also provides a velocity tracking inner loop. The robust term deals with unstructured unmodeled dynamics and disturbances. However, the robust NN controller can control the velocity tracking error so that it converges to zero, such that the stable state error also converges to zero.

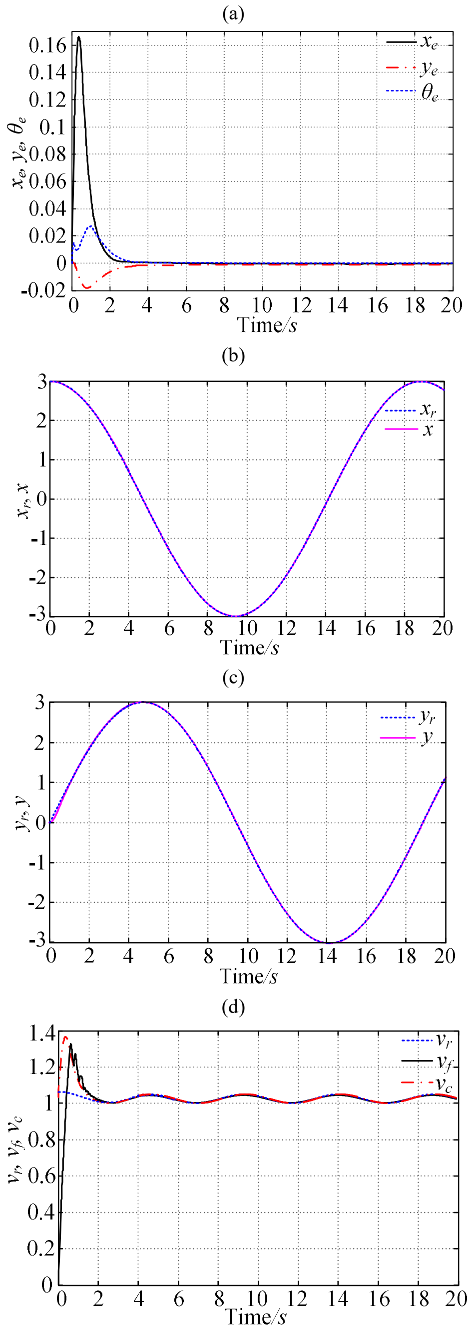

Fig. 5. No slippage with the NN controller:
(a) Position errors (b) Position error in the *x*-axis
(c) Position error in the *y*-axis (d) Velocity tracking

5.2 Robust NN controller with slippage

The response of this controller is described based on the tangential slip of the wheels, as shown in Fig. 6. The results show that tracking errors occurred and there were large position and velocity error changes. This controller was unstable.

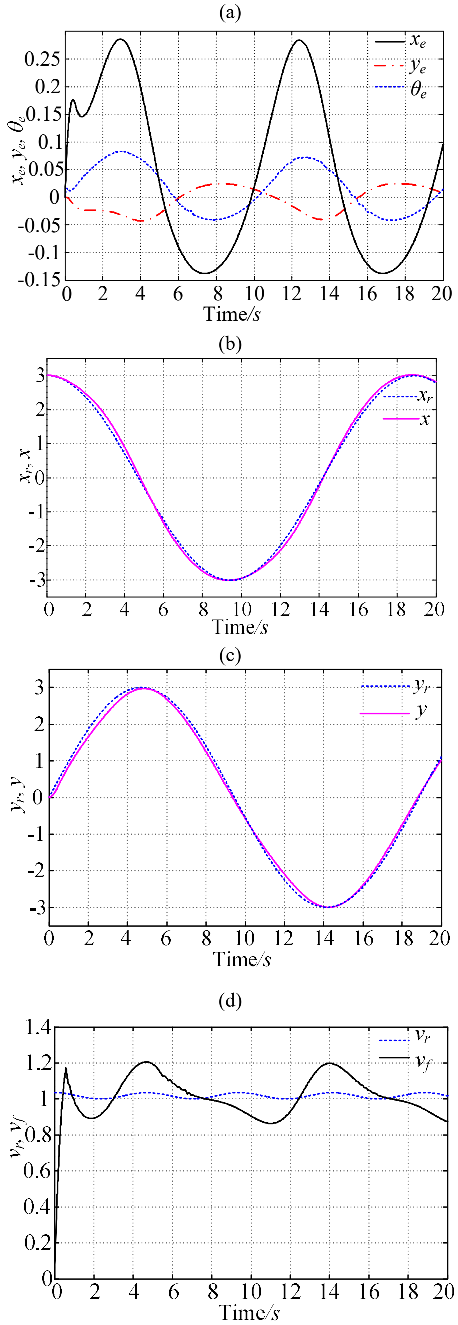

Fig. 6. Existing slippage with the NN controller:
(a) Position errors (b) Position error in X-axis
(c) Position error in Y-axis (d) Velocity tracking

5.3 Robust NN controller with slip-compensation

The response of this controller is shown in Fig.7. We compensated for lost velocity due to wheel slippage. It is obvious that the tracking errors were almost the same as A. The results show that slip-compensation plays a significant role. Thus, when slippage occurred between the wheels and the ground, the control method with slip-compensation and adaptive NN control laws performed well during tracking control.

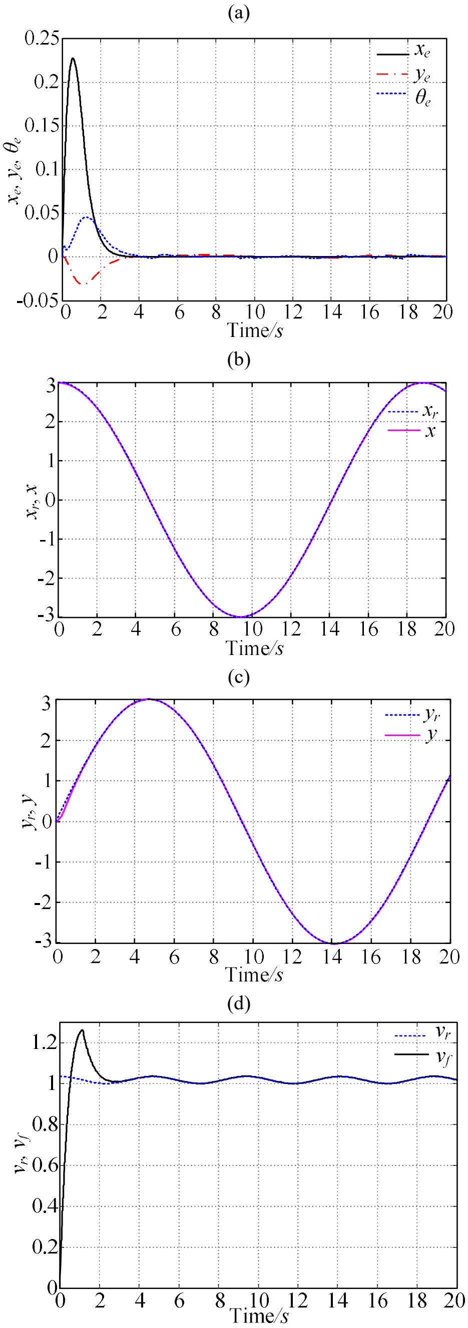

Fig. 7. No slippage with the NN controller:
(a) Position errors (b) Position error in X-axis
(c) Position error in Y-axis (d) Velocity tracking

6. Conclusion

In this paper, we proposed and demonstrated a robust adaptive motion tracking control method based on RBF neural networks and slip-compensation. This control scheme was designed for tracking the desired motion trajectory of a wheeled mobile robot system. We also designed a neural network learning procedure that enhanced the performance of the proposed control scheme, even when tracking on uneven terrain. The stability of the inner closed-loop system was analyzed. We showed that the velocity tracking error converges to zero and that the convergence of the position tracking errors to zero is guaranteed by the proposed robust adaptive neural network motion tracking control law (39), even though external disturbances and unknown system parameters, such as friction and slippage, are very difficult to model using conventional techniques. To overcome this difficulty, we compensated for the slip ratios of all the wheels and we derived an RBF neural network controller with guaranteed performance and described its advantages. One of the most important advantages of the control methodology is that no prior knowledge is required for the system parameters or for the thresholds and weights of the neural networks. Finally, we demonstrated the results of precise tracking performance using a Matlab simulation.

Acknowledgements

This work is supported by the National Natural Science Foundation (NNSF) of China (No.51605393) and the China Postdoctoral Science Foundation (No.2018M633398).

References

B. M. Kim, and P. Tsiotras, "Controllers for Unicycle-Type Wheeled Robots: Theoretical Results and Experimental Validation," IEEE Transactions on Robotics and Automation, vol. 18, pp. 294-307, June. 2002.

Y. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," Proc. IEEE International Conference on Robotics and Automation, vol. 1, pp. 384-389, 1990.

D. H. Kim and J. H. Oh, "Tracking control of a two-wheeled mobile robot using input-output linearization," Control Engineering Practice, vol. 7, pp. 369-373, 1999.

N. R. Julio et al, "Mobile robot path tracking using a robust PID controller," Control Engineering Practice, vol. 9, pp. 1209-1214, 2001.

K. D. Do, and J. Pan, "Global output-feedback path tracking of unicycle- type mobile robots," Robotics and Computer-Integrated Manufacturing, vol. 22, pp. 166-179, 2006.

Z. P. Wang, S. S. Ge, T. H. Lee, and X. C. Lai, "Adaptive Smart Neural Network Tracking Control of Wheeled Mobile Robots," ICARCV, 2006.

Z. J. Li, and C. Q. Xu, "Adaptive fuzzy logic control of dynamic balance and motion for wheeled inverted pendulums," Fuzzy Sets and Systems, vol. 160, no. 12, pp. 1787-1803, 2009.

R. Fierro, and F. L. Lewis, "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics," Journal of Robotic Systems, vol. 14, no. 3, pp. 149-163, 1997.

L. R. Ray, D. C. Brande, and J. H. Lever, "Estimation of net traction for differential-steered wheeled robots," Journal of Terramechanics, vol. 46, pp. 75-87, 2009.

J. Y. Wong, Theory of Ground Vehicles, John Wiley & Sons, 1978.

L. Ding, H. B. Gao, Z. Q. Deng, and Z. Liu, "Slip-Ratio-Coordinated Control of Planetary Exploration Robots Traversing over Deformable Rough Terrain," IEEE International Conference on Intelligent Robots and Systems, vol. 10, pp. 4958-4963, 2010.

L. Ding, H. B. Gao, Z. Q. Deng etc, "Experimental study and analysis on driving wheels' performance for planetary exploration rovers moving in deformable soil," Journal of Terramechanics, vol. 48, pp. 27-45, 2011.

D. Endo, Y. Okada, K. Nagatani, and K. Yoshida, "Path Following Control for Tracked Vehicles Based on Slip-Compensating Odometry," Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems San Diego, CA, USA, Oct 29-Nov 2, pp. 2871-2876, 2007.

Z. J. Li, Y. P. Yang, and J. X. Li, "Adaptive Motion/Force Control of Mobile Under-actuated Manipulators with Dynamics Uncertainties by Dynamic Coupling and Output Feedback," IEEE Trans. Control System Technology, vol. 5, no. 18, pp. 1068-1079, 2010.

F. L. Lewis, C. T. Abdallah, and D. M. Dawson, "Control of Robot Manipulators," New York: MacMillan, 1993.

N. B. Karayiannis, G. W. Mi, G. R. Basis, "Merging Supervised and Unsupervised Learning with Network Growth Techniques," IEEE Transactions on Neural Networks, vol. 6, no. 8, pp. 339-356, 1997.

J. R. Jung, B. J. Yum, "Artificial neural network based approach for dynamic parameter design," Expert System with Applications, vol. 38, pp. 504-510, 2011.

Gao, X., et al., Adaptive High Order Neural Network Identification and Control for Hysteresis Nonlinear System with Bouc-Wen Model. International Journal of Applied Mathematics in Control Engineering, 2019. 2(1): p. 17-24.

R. A. Horn, and C. R. Johnson, "Matrix Analysis," New York: Cambridge Univ. Press, 1985.

S. Haykin, "Neural Networks and Learning Machines," Third Edition, 2009.

Wu, W.J., Parametric Lyapunov Equation Approach to Stabilization and Set Invariance Conditions of Linear Systems with Input Saturation. International Journal of Applied Mathematics in Control Engineering, 2018. 1(1): p. 79-84.

Song Xingguo is received the Ph.D. degree in mechanical engineering from Harbin Institute of Technology in 2015. He is currently a lecturer with the School of Mechanical Engineering, Southwest Jiaotong University. His main research interests are in the areas of control theory and Robotics.

Liu Tingting received the M.S. degree in Chinese literature from Heilongjiang University in 2014. She is currently a Part-time financial staff with the School of Mechanical Engineering, Southwest Jiaotong University. Her main research interests are in the areas of Chinese-English translation.