International Journal of Control, 2014
Vol. 87, No. 8, 1513–1522, http://dx.doi.org/10.1080/00207179.2013.878038

Adaptive motion control of wheeled mobile robot with unknown slippage

Haibo Gao, Xingguo Song*, Liang Ding, Kerui Xia, Nan Li and Zongquan Deng

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, China

(*Received 3 March 2013; accepted 18 December 2013*)

As a major representative nonholonomic system, wheeled mobile robot (WMR) is often used to travel across off-road environments that could be unstructured environments. Slippage often occurs when WMR moves in slopes or uneven terrain, and the slippage generates large accumulated position 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 adaptive controller to allow WMR to track the desired trajectory under unknown longitudinal slip, where the stabilisation of the closed-loop tracking system is guaranteed by the Lyapunov theory. All system states use neural network online weight tuning algorithms, which ensure 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.

Keywords: nonholonomic systems; wheeled mobile robot (WMR); radial basis function (RBF); neural networks (NN); slip ratio; Lyapunov theory

1. Introduction

In the past decades, several researchers have investigated the problem of controlling nonholonomic systems. As a major representative nonholonomic system, the wheeled mobile robot (WMR) has attracted considerable attention. Most control methods for WMR are based on this assumption that the wheels roll without slipping.

Under this assumption, many modern control theories and algorithms have been proposed for the motion control of WMR. A classical stable control scheme was proposed (Kanayama, Kimura, Miyazaki, & Noguchi, 1990) to compute the vehicle control inputs for an autonomous mobile robot under the assumption of perfect velocity tracking. This kind of control method solved the nonholonomic tracking problems based on over-simplified models in the initial periods of their development by neglecting the vehicle dynamics and thus considering only the steering system, or considering a known mathematical model of dynamical systems. Thereby, many robust nonlinear and adaptive control methods (Ge, Wang, Lee, & Zhou, 2001; Li & Xu, 2009; Li, Li, & Kang, 2010) were designed to control the motion of mobile robots, these control performances are often degraded by modelling errors, information feedback errors, and external disturbances. However, there still exist some sorts of uncertainties in the dynamics of nonholonomic systems or environmental information, and their influences are very important in practice. This means that we need a more intelligent controller to solve these problems.

Fully autonomous WMR control systems need to cope with dynamical robot uncertainties, unmodelled or unstructured disturbances, and nonlinear friction. But, it is difficult to obtain an accurate mathematical model for applying computed torque controllers or other model-based controllers in practice. Following the neural network (NN) development, the neural network-based control of mobile robots has been the subject of intense research in recent years (Fierro & Lewis, 1998; Jolly, Kumar, & Vijayakumar, 2009; Sun, Pei, Pan, & Zhang, 2013; Wang, Ge, Lee, & Lai, 2006). These researches had produced new methods for solving the main difficulties. A neural network-based model that combined the backstepping technique with a torque controller was presented by Fierro and Lewis (1998), and this algorithm was based on the application of multi-layered BP neural networks. The tracking control using an adaptive smart neural network for WMR was investigated (Wang et al. 2006), and it produced fine motion control based on partially unknown dynamics. Sun et al. (2013) proposed a robust adaptive NN control for the nonholonomic mobile robot to track the desired environmental boundary. These control methods stated above are designed under the constraint of pure roll and no slip.

However, we know the applications of WMR usually require them to travel across off-road environments in task, such that some phenomena as slippage or sliding always exists between the wheels and ground, which cause rolling of wheels, are not perfect (Wong, 2001). When a WMR

* Corresponding author. Email: xg.song@hotmail.com

moves on uneven terrain or slopes, the slippage will generate large numbers of accumulate position errors in the vehicle compared with conventional WMR (Ding et al., 2011). Therefore, the tracking control of mobile robot will be considerably influenced by the condition of terrain (Ray, Brande, & Lever, 2009). In order to solve this problem, we propose an approach based on the estimation of each wheel's quantitative slip, and incorporate the slip ratio parameters into the kinematic model of the WMR. Many researches have addressed the slip phenomenon in the WMR (Endo, Okada, Nagatani, & Yoshida, 2007; Moosavian & Kalantari, 2008). The slip ratios of all wheels could be estimated by an experimental study (Ding, Gao, Deng, & Liu, 2010; Iossaqui, Camino, & Zampieri, 2010). These methods proved useful for estimating the slip ratios of wheels.

In this paper, we propose an effective adaptive motion tracking control method based on NN and slip-compensation for WMR systems. The control objective is to track a specified motion trajectory under the slip occurrence. We calculate the wheel-slips, and develop the proposed control method using NN to model the system dynamics and nonlinearities. The radial basis function (RBF) of neural networks is well suited to uncertain or nonlinear functions because of its rapid online learning ability and nonlinear characteristics. We use the control approach to overcome unknown system parameters and slippage in the WMR system, and to find a suitable control input that stabilises the closed-loop system. This controller guarantees perfect velocity tracking and the posture error converges to minimum. The proposed control method facilitates precise motion tracking performance, which was demonstrated using the Matlab simulation.

The remainder of this paper is organised as follows. The basics of nonholonomic system and slip ratios, and RBF neural networks are introduced in Section 2. Section 3 discusses the adaptive control method combined with neural networks and slip-compensation, and Section 4 covers the stability analysis. The Matlab simulation is presented in Section 5. Finally, Section 6 concludes the paper.

2. Problem formulation

2.1 *A nonholonomic WMR model*

A mobile robot system in an n-dimensional configuration space $\boldsymbol{C}$ with the generalised coordinates $(q_1, \ldots, q_n)$ that is subject to m constraints can be described by Fierro and Lewis (1998):

$$\begin{aligned}&\boldsymbol{M}(q)\ddot{q} + \boldsymbol{V}(q,\dot{q})\dot{q} + \boldsymbol{F}(\dot{q}) + \boldsymbol{G}(q) + \boldsymbol{\tau}_d \\ &\quad = \boldsymbol{B}(q)\boldsymbol{\tau} - \boldsymbol{A}^{\mathrm{T}}(q)\lambda,\end{aligned} \tag{1}$$

where $\boldsymbol{M} \in R^{n\times n}$ is a symmetric, positive definite inertia matrix, $\boldsymbol{V} \in R^{n\times n}$ denotes the centripetal and coriolis forces, $\boldsymbol{F} \in R^{n\times 1}$ denotes the surface friction, $\boldsymbol{G} \in R^{n\times 1}$ is gravitational torques (or forces), $\boldsymbol{\tau}_d \in R^{n\times 1}$ denotes bounded unknown disturbances, including unstructured unmodelled dynamics, $\boldsymbol{B} \in R^{n\times r}$ is the input transformation matrix, $\boldsymbol{\tau}_d \in R^{r\times 1}$ is the input vector of forces and torques supplied by actuators, $\boldsymbol{A} \in R^{m\times n}$ is the matrix associated with the constraints, and $\lambda \in R^{m\times 1}$ is the vector of constraint forces.

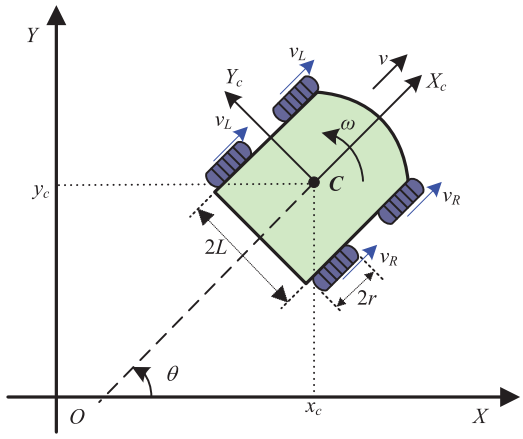


Figure 1. A nonholonomic wheeled mobile robot.

In this paper, we use four-wheeled mobile robot as an example. The mobile robot shown in Figure 1 is a typical example of a nonholonomic mechanical system, and skid-steered autonomous WMR. The motion and orientation are controlled by independent actuators, i.e., DC motors provide the necessary torque to the wheels. 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 centre $\boldsymbol{C}$ 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. Due to the symmetrical property of the platform, we assume $\boldsymbol{C}$ is also the geometrical centre.

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 (Fierro & Lewis, 1998):

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

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

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

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

$$\boldsymbol{A}(q)\boldsymbol{S}(q) = 0. \tag{3}$$

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

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

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

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

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

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \tag{6}$$

where $v = [v \quad \omega]^{\mathrm{T}}$, v and ω are the displacement/linear and the angular velocities of the mass centre of the vehicle body, respectively. In addition, $|v| \leq v_{\max}$ and $|\omega| \leq \omega_{\max}$, $v_{\max}$ 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 can rewrite v and ω to reflect the effects of the slippage:

$$v = \frac{v_R(1-s_1) + v_L(1-s_2)}{2}, \tag{7}$$

$$\omega = \frac{v_R(1-s_1) - v_L(1-s_2)}{2L}, \tag{8}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1-s_1 & 1-s_2 \\ \dfrac{1-s_1}{L} & \dfrac{1-s_2}{-L} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix}, \tag{9}$$

where vector $\boldsymbol{v} = [v_R v_L]^{\mathrm{T}}$, v_R and v_L are the driving velocities of the right and left wheels, respectively, which can be measured by encoders. $2L$ is the right- and left-wheel tread, and s_1 and s_2 denote the slip ratios defined as follows:

$$s_1 = \frac{v_R - v_R'}{v_R}, s_2 = \frac{v_L - v_L'}{v_L}, \tag{10}$$

where v_R' and v_L' are the factual/current velocity of the right and left wheels, respectively, while s_1 and s_2 are the slip ratios of the right and left wheels, respectively. If no slippage occurs, the current velocity is equal to the driving velocity ($v_R' = v_R$, $v_L' = v_L$), and slip ratios $s_1 = s_2 = 0$.

Substituting (9) into (6), we can rewrite the kinematic equation by

$$\dot{q} = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (1-s_1)c\theta & (1-s_2)c\theta \\ (1-s_1)s\theta & (1-s_2)s\theta \\ 1-s_1/L & 1-s_2/L \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix}, \tag{11}$$

where $c\theta = \cos\theta$, $s\theta = \sin\theta$,

$$\bar{S} = \frac{1}{2} \begin{bmatrix} (1-s_1)c\theta & (1-s_2)c\theta \\ (1-s_1)s\theta & (1-s_2)s\theta \\ 1-s_1/L & 1-s_2/L \end{bmatrix}. \tag{12}$$

According to (11), we can obtain the slip ratios of the right and left wheels as follows:

$$s_1 = 1 - \frac{\dot{x}_c + \dot{\theta}Lc\theta}{v_R c\theta}, s_2 = 1 - \frac{\dot{x}_c - \dot{\theta}Lc\theta}{v_L c\theta}. \tag{13}$$

Slip ratios are determined by physical interactions between the wheels and the ground; however, in general, the aspects of wheel-terrain interaction that are needed for accurate models are neither well known nor easily measurable in realistic situations. Therefore, we assume that the current angular velocity of the vehicle body, ω, can be directly measured by a gyro-sensor. The driving velocities of wheels, v_R and v_L can be detected by using encoders. Then, odometry can be implemented with consideration of the slip, to detect slip ratios of both wheels using model (13), quantitatively.

2.2 *Structural properties of a mobile robot*

The dynamics of system (1) is now transformed into a more appropriate representation for control purposes. Using Equation (12), the dynamics Equation (1) can be rewritten by (Fierro & Lewis, 1998):

$$\begin{aligned} &\dot{q} = \bar{\boldsymbol{S}}(q)\boldsymbol{v}(t) \\ &\bar{\boldsymbol{M}}(q)\dot{\boldsymbol{v}} + \bar{\boldsymbol{V}}(q,\dot{q})\boldsymbol{v} + \bar{\boldsymbol{F}}(\boldsymbol{v}) + \bar{\boldsymbol{G}}(\boldsymbol{v}) + \bar{\boldsymbol{\tau}}_d = \bar{\boldsymbol{\tau}}, \end{aligned} \tag{14}$$

where $\bar{\boldsymbol{M}} = \bar{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{M}\bar{\boldsymbol{S}}$, $\bar{\boldsymbol{V}} = \bar{\boldsymbol{S}}^{\mathrm{T}}(\boldsymbol{M}\dot{\bar{\boldsymbol{S}}} + \boldsymbol{V}\bar{\boldsymbol{S}})$, $\bar{\boldsymbol{F}} = \bar{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{F}$, $\bar{\boldsymbol{G}} = \bar{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{G}$, $\bar{\boldsymbol{\tau}}_d = \bar{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{\tau}_d$, $\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{B}\tau$ and $\|\bar{\boldsymbol{\tau}}_d\| \leq b_d$ with b_d as an upper bound of system disturbances.

Property 1: $\bar{\boldsymbol{M}}$ is a symmetric positive definite, and the matrix $\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{V}}$ is skew-symmetric (Fierro & Lewis, 1998): i.e.,

$$x^{\mathrm{T}}(\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{V}})x = 0, \forall x \neq 0.$$

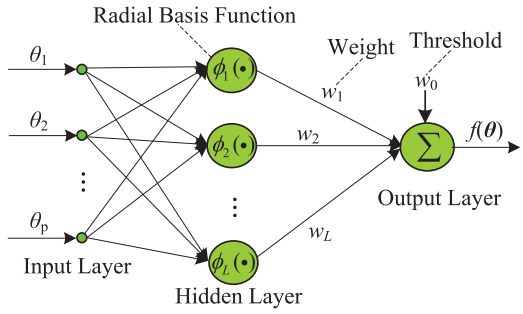


Figure 2. Structure of RBF neural network

2.3 *RBF neural networks*

Many different models of neural networks have been established (Er & Gao, 2003; Jung & Yum, 2011; Li & Yang, 2012; Liaw, Farrell, & Schaal, 2009; Theodoridis, Boutalis, & Christodoulou, 2011) for specific objectives. Of these models, we find the RBF neural network is well suited to modelling uncertain or nonlinear functions. A typical RBF neural network is shown in Figure 2. It is a two-layer network composed of a hidden layer and an output layer. Using this simple structure, the RBF neural network facilitates a more effective weight updating procedure compared with other, more complex multilayer networks.

A schematic for the RBF network is illustrated in Figure 2. In this paper, the Gaussian type function is considered as

$$\phi_i(\boldsymbol{\theta}) = \exp\left[-\frac{1}{2\sigma_i^2}\|\boldsymbol{\theta} - \mu_i\|^2\right], \tag{15}$$

where μ_i, σ_i are the centre and width of the ith neuron, $i = 1, \ldots, L$ in (15). The activation of a neuron in the output layer is determined by a linear combination of the fixed nonlinear basis functions, i.e.

$$g(\theta) = \sum_{i=1}^{m} w_i \phi_i(\theta),$$

where w_i are the adjustable weights that link the output nodes with the appropriate hidden neurons. These weights in the output layer can be learnt using the least-squares method.

In this study, the RBF neural network is employed for nonlinear function approximation. We assume a special case of the linear regression model as follows:

$$g(\boldsymbol{\theta}) = \boldsymbol{W}\phi(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta}), \tag{16}$$

$$\boldsymbol{W} = [w_0, w_1, \ldots, w_L],$$

$$\phi(\boldsymbol{\theta}) = [1, \phi_1(\boldsymbol{\theta}), \ldots, \phi_L(\boldsymbol{\theta})]^{\mathrm{T}}, \tag{17}$$

where $\boldsymbol{\theta} \in R^p$ is the input vector, $\boldsymbol{W} \in R^{L+1}$ contains the ideal threshold w_0, and weights $w_1 \cdots w_L$ of the neural network, $\phi(\boldsymbol{\theta}) \in R^{(L+1)}$ is the activation vector comprising the RBF, and $\varepsilon(\boldsymbol{\theta}) \in R$ is the neural network function approximation error, which is assumed to be uncorrelated with the regressor $\phi(\boldsymbol{\theta})$.

Using the neural network shown in Figure 2 with a sufficiently large number (L) of RBF in the hidden layer to approximate the smooth function $g(\boldsymbol{\theta})$ described by (9), there exists a positive number b_f, such that $|\varepsilon(\boldsymbol{\theta})| \leq b_f, \forall \boldsymbol{\theta}$. Then, an approximation of $g(\boldsymbol{\theta})$ can be given by

$$\hat{g}(\boldsymbol{\theta}) = \hat{W}^{\mathrm{T}}\phi(\boldsymbol{\theta}), \tag{18}$$

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

Here, the NN is trained to match specified exemplar pairs $(\boldsymbol{\theta}, g(\boldsymbol{\theta}))$, with $\boldsymbol{\theta}$ the ideal training system states. The corresponding actual output is $\hat{g}(\boldsymbol{\theta})$, while the target output is $g(\boldsymbol{\theta})$. For RBF neural networks, a common weight tuning algorithm is the gradient algorithm based on the *backpropagated* error (Haykin, 2009).

3. Adaptive tracking controller design

3.1 *Motion tracking problem*

In this control system, we use two postures: a *reference posture* $q_r = (x_r, y_r, \theta_r)^{\mathrm{T}}$ that is a goal posture of the vehicle, and a *current posture* $q = (x_c, y_c, \theta)^{\mathrm{T}}$ that 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 (Kanayama et al., 1990) as:

$$\boldsymbol{e}_m = \begin{bmatrix} x_e & y_e & \theta_e \end{bmatrix}^{\mathrm{T}} = \boldsymbol{T}_e \boldsymbol{e}, \tag{19}$$

where

$$\boldsymbol{T}_e = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \boldsymbol{e} = \begin{bmatrix} x_r - x_c \\ y_r - y_c \\ \theta_r - \theta \end{bmatrix}.$$

An auxiliary velocity control law input that achieves tracking for (3) is given by (Kanayama et al., 1990):

$$\boldsymbol{v}_c = \begin{bmatrix} v_r \cos\theta_e + k_1 x_e \\ \omega_r + k_2 v_r y_e + k_3 v_r \sin\theta_e \end{bmatrix} \tag{20}$$

where $(k_1, k_2, k_2) > 0$ are the feedback gains of x_e, y_e, and θ_e, respectively.

However, 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, according to (7), (8) and (20), based on the estimation of slip ratios, we can recover an auxiliary

velocity control law input $\boldsymbol{v}_c = [v_1, v_2]^{\mathrm{T}}$ by

$$\boldsymbol{v}_c = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \frac{v_c + \omega_c L}{1-\hat{s}_1} \\ \frac{v_c - \omega_c L}{1-\hat{s}_2} \end{bmatrix}, \tag{21}$$

where v_1 and v_2 are the auxiliary velocity control law input of the right and left wheels, respectively; $\hat{s}_1$ and $\hat{s}_2$ are the estimated slip ratios of the right and left wheels, which can be detected by equality (13). From literature (Ding et al., 2010), we know slip ratio is defined by $s \in [-1, 1]$, when wheels are slipping on original location and not able to move forward/backward, in this case, slip ratio $s = 1$ or -1. Therefore, for tracking object, we should assume $\hat{s} \neq \pm 1$.

After slip-compensation, the velocity tracking errors of the wheels are defined as follows:

$$\boldsymbol{e}_c = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \boldsymbol{v}_c - \boldsymbol{v}. \tag{22}$$

Differentiating (22), using (14), the mobile robot dynamics may be written in terms of the tracking errors as

$$\bar{\boldsymbol{M}}(q)\dot{\boldsymbol{e}}_c = -\bar{\boldsymbol{V}}(q, \dot{q})\boldsymbol{e}_c - \bar{\boldsymbol{\tau}} + g(\boldsymbol{x}) + \bar{\boldsymbol{\tau}}_d, \tag{23}$$

where the important nonlinear mobile robot function is defined as

$$g(\boldsymbol{x}) = \bar{\boldsymbol{M}}\dot{\boldsymbol{v}}_c + \bar{\boldsymbol{V}}(q, \dot{q})\boldsymbol{v}_c + \bar{\boldsymbol{F}}(\boldsymbol{v}) + \bar{\boldsymbol{G}}(\boldsymbol{v}). \tag{24}$$

Here, the vector $\boldsymbol{x}$ required to compute $g(\boldsymbol{x})$ can be defined as

$$\boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{v}_c^{\mathrm{T}} & \dot{\boldsymbol{v}}_c^{\mathrm{T}} & \boldsymbol{v}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}. \tag{25}$$

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

3.2 *Adaptive NN control scheme*

In this paper, the unknown system function $g(\boldsymbol{x})$ is approximated using the RBF neural network described by (15). A major advantage is that this can always be accomplished, due to the RBF neural network approximation property. To consider the function $g(\boldsymbol{x})$ given by (24), the vector in (15) is defined as:

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

The neural network (15) for (24) is redefined as $g(\boldsymbol{\theta})$: $R^6 \rightarrow R^2$, which is rewritten as:

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

where $\boldsymbol{w} \in R^{(L+1)\times 2}$ is the vector of the ideal threshold and their weights. The bounds for w and $\varepsilon(\boldsymbol{\theta})$ are expressed as

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

Then, an estimate of $g(\boldsymbol{\theta})$ can be given by

$$\hat{g}(\boldsymbol{\theta}) = \hat{\boldsymbol{w}}^{\mathrm{T}}\phi(\boldsymbol{\theta}) \tag{29}$$

where $\hat{\boldsymbol{w}} \in R^{(L+1)\times 2}$ is the vector of the estimated threshold and weights.

The control law is given by

$$\bar{\boldsymbol{\tau}} = \hat{g} + \boldsymbol{K}\boldsymbol{e}_c + l\,\mathrm{sgn}(\boldsymbol{e}_c), \tag{30}$$

where $\hat{g}$ is regarded as the output torque of NN controller; $\boldsymbol{K}\boldsymbol{e}_c$ is the torque controller, and, $\boldsymbol{K} = diag[k_4, k_5]$ is a positive definite matrix; $l\,\mathrm{sgn}(\boldsymbol{e}_c)$ is a robust term to suppress the effect of disturbance and approximate errors.

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

$$\dot{\hat{w}}_i = \phi \boldsymbol{e}_c^{\mathrm{T}} - \kappa \|\boldsymbol{e}_c\| \hat{w}_i, i = 1, 2, \ldots, L, \tag{31}$$

where $\hat{\boldsymbol{w}} = [\hat{w}_1 \ldots \hat{w}_L]$, L is the number of NN hidden layer, and learning law κ is a positive constant. The simultaneous updates of the adaptive law may be suitable for non-stationary conditions or online settings.

The control law (30) can be rewritten as:

$$\bar{\boldsymbol{\tau}} = \hat{\boldsymbol{w}}^{\mathrm{T}}\phi(\boldsymbol{\theta}) + \boldsymbol{K}\boldsymbol{e}_c + l\,\mathrm{sgn}(\boldsymbol{e}_c) \tag{32}$$

and the parameter l is defined as

$$l \geq b_\varepsilon + b_d + \frac{1}{4}\kappa b_w^2 + \epsilon, \tag{33}$$

which is related to the bounds described by (28), the parameter κ in (31), and a strictly positive constant ε.

4. Stability analysis

In this section, we perform system stability analysis of the closed-loop behaviour in the proposed control method. Thus, we derive and analyse the closed-loop dynamics. Substituting the control input (32) into the mobile robot dynamics system described by (23) yields:

$$\bar{\boldsymbol{M}}\dot{\boldsymbol{e}}_c = -(\boldsymbol{K} + \bar{V})\boldsymbol{e}_c + \tilde{g} + \bar{\tau}_d - l\,\mathrm{sgn}(\boldsymbol{e}_c), \tag{34}$$

where $\tilde{g} = g - \hat{g}$ is the function estimation error. This estimation error is expressed according to (27) and (29) as

$$\tilde{g}(\boldsymbol{\theta}) = \tilde{\boldsymbol{w}}^{\mathrm{T}}\phi(\boldsymbol{\theta}) + \varepsilon(\boldsymbol{\theta}), \tag{35}$$

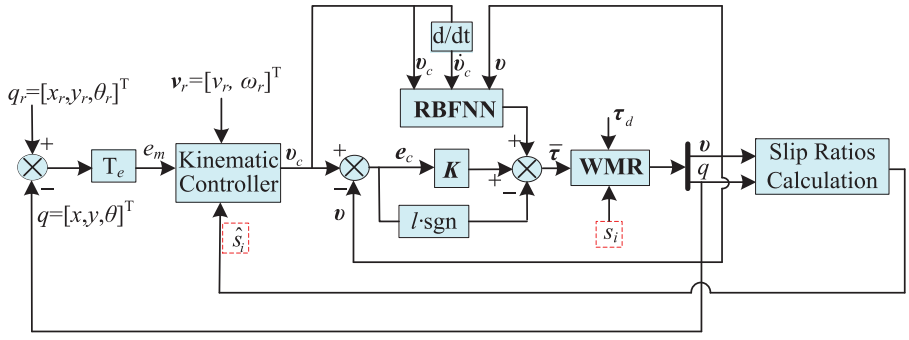


Figure 3. Structure of the proposed motion control method.

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, (33) can be written as follows:

$$\begin{aligned}\bar{\boldsymbol{M}}\dot{\boldsymbol{e}}_c = &-(\boldsymbol{K}+\bar{\boldsymbol{V}})\boldsymbol{e}_c + \tilde{\boldsymbol{w}}^{\mathrm{T}}\phi(\boldsymbol{\theta}) \\ &+\varepsilon(\boldsymbol{\theta}) + \bar{\boldsymbol{\tau}}_d - l\,\mathrm{sgn}(\boldsymbol{e}_c).\end{aligned} \tag{36}$$

It is necessary to develop the adaptive NN control (NNC) given in Equation (32) by considering the system stability, so that both the position and the velocity tracking errors converge to zero. Consider the following Lyapunov functional candidate for the proposed controller:

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

and

$$V_1(\boldsymbol{e}_c, t) = \frac{1}{2}\boldsymbol{e}_c^{\mathrm{T}}\bar{\boldsymbol{M}}\boldsymbol{e}_c + \frac{1}{2}[(\boldsymbol{w}-\hat{\boldsymbol{w}})^{\mathrm{T}}(\boldsymbol{w}-\hat{\boldsymbol{w}})], \tag{37}$$

where matrix $\bar{\boldsymbol{M}}$ is a symmetric positive definite, so that $V_1 \geq 0$. Differentiating Equation (37) with respect to time as

$$\begin{aligned}&\dot{V}_1(\boldsymbol{e}_c, t) \\ &= \boldsymbol{e}_c^{\mathrm{T}}\bar{\boldsymbol{M}}\dot{\boldsymbol{e}}_c + \frac{1}{2}\boldsymbol{e}_c^{\mathrm{T}}\dot{\bar{\boldsymbol{M}}}\boldsymbol{e}_c + \mathrm{tr}[(\boldsymbol{w}-\hat{\boldsymbol{w}})^T\dot{\hat{\boldsymbol{w}}}] \\ &= \boldsymbol{e}_c^{\mathrm{T}}(-\bar{\boldsymbol{V}}\boldsymbol{e}_c + \tilde{\boldsymbol{w}}^{\mathrm{T}}\phi - \boldsymbol{K}\boldsymbol{e}_c - l\,\mathrm{sgn}(\boldsymbol{e}_c) + \varepsilon + \bar{\boldsymbol{\tau}}_d) \\ &\quad + \frac{1}{2}\boldsymbol{e}_c^{\mathrm{T}}\dot{\bar{\boldsymbol{M}}}\boldsymbol{e}_c - \mathrm{tr}\left[\tilde{\boldsymbol{w}}^{\mathrm{T}}\left(\phi\boldsymbol{e}_c^{\mathrm{T}} - \kappa\,\|\boldsymbol{e}_c\|\,\hat{\boldsymbol{w}}\right)\right] \\ &= \frac{1}{2}\boldsymbol{e}_c^{\mathrm{T}}(\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{V}})\boldsymbol{e}_c + \boldsymbol{e}_c^{\mathrm{T}}\tilde{\boldsymbol{w}}^{\mathrm{T}}\phi - \boldsymbol{e}_c^{\mathrm{T}}\boldsymbol{K}\boldsymbol{e}_c \\ &\quad - l\boldsymbol{e}_c^{\mathrm{T}}\mathrm{sgn}(\boldsymbol{e}_c) + \boldsymbol{e}_c^{\mathrm{T}}(\boldsymbol{\varepsilon} + \bar{\boldsymbol{\tau}}_d) \\ &\quad - \mathrm{tr}(\tilde{\boldsymbol{w}}^{\mathrm{T}}\phi\boldsymbol{e}_c^{\mathrm{T}} - \kappa\tilde{\boldsymbol{w}}^{\mathrm{T}}\,\|\boldsymbol{e}_c\|\,\|\hat{\boldsymbol{w}}\|) \\ &= \boldsymbol{e}_c^{\mathrm{T}}\tilde{\boldsymbol{w}}^{\mathrm{T}}\phi - \boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - l\,\|\boldsymbol{e}_c\| + \boldsymbol{e}_c^{\mathrm{T}}(\boldsymbol{\varepsilon} + \bar{\tau}_d) \\ &\quad - \mathrm{tr}(\boldsymbol{e}_c^{\mathrm{T}}\tilde{\boldsymbol{w}}^{\mathrm{T}}\phi) + \mathrm{tr}[\kappa\tilde{\boldsymbol{w}}\,\|\boldsymbol{e}_c\|\,(\boldsymbol{w}-\tilde{\boldsymbol{w}})] \\ &\leq -\boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - l\,\|\boldsymbol{e}_c\| + \|\boldsymbol{e}_c\|\,(\boldsymbol{\varepsilon} + \bar{\tau}_d) \\ &\quad - \kappa\,\|\boldsymbol{e}_c\|\,\|\tilde{\boldsymbol{w}}\|^2 + \kappa\,\|\boldsymbol{e}_c\|\,\|\tilde{\boldsymbol{w}}\|\,b_w \\ &= -\boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - l\,\|\boldsymbol{e}_c\| + \|\boldsymbol{e}_c\|\,(b_\varepsilon + b_d) \\ &\quad - \kappa\,\|\boldsymbol{e}_c\|\,(\|\tilde{\boldsymbol{w}}\|^2 - \|\tilde{\boldsymbol{w}}\|\,b_w) \\ &\quad - \frac{1}{4}\kappa b_w^2\,\|\boldsymbol{e}_c\| + \frac{1}{4}\kappa b_w^2\,\|\boldsymbol{e}_c\| \\ &= -\boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - l\,\|\boldsymbol{e}_c\| + \|\boldsymbol{e}_c\|\,(b_\varepsilon + b_d) \\ &\quad - \kappa\,\|\boldsymbol{e}_c\|\,(\|\tilde{\boldsymbol{w}}\|^2 - \|\tilde{\boldsymbol{w}}\|\,b_w + \frac{1}{4}b_w^2) + \frac{1}{4}\kappa b_w^2\,\|\boldsymbol{e}_c\| \\ &= -\boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - \|\boldsymbol{e}_c\|\left\{l - b_\varepsilon - b_d\right. \\ &\quad \left. + \kappa\left[\left(\|\tilde{\boldsymbol{w}}\| - \frac{1}{2}b_w\right)^2 - \frac{1}{4}b_w^2\right]\right\} \\ &\leq -\boldsymbol{K}\,\|\boldsymbol{e}_c\|^2 - \|\boldsymbol{e}_c\|\left[\epsilon + \kappa\left(\|\tilde{\boldsymbol{w}}\| - \frac{1}{2}b_w)^2\right)\right],\end{aligned} \tag{38}$$

where inequality (33) and property 2, $x^{\mathrm{T}}(\dot{\bar{\boldsymbol{M}}} - 2\bar{\boldsymbol{V}})x = 0$, $\forall x \neq 0$ are used, and tr[] denotes the trace of matrix (Horn & Johnson, 1985).

With (38), $\dot{V}_1 \leq 0$ is guaranteed to be negative. This shows $V_1 \to 0$ and implies $\boldsymbol{e}_c(t) \to 0$ and $\dot{\boldsymbol{e}}_c(t) \to 0$ as $t \to \infty$. The tracking velocity convergence is guaranteed by the control law (32), which implies $\boldsymbol{v} \to \boldsymbol{v}_c$ and $\dot{\boldsymbol{v}} \to \dot{\boldsymbol{v}}_c$ as $t \to \infty$.

We consider the other positive definite Lyapunov functional candidate:

$$V_2(\boldsymbol{e}_m, t) = \frac{1}{2}(x_e^2 + y_e^2) + (1 - \cos\theta_e)/k_2. \tag{39}$$

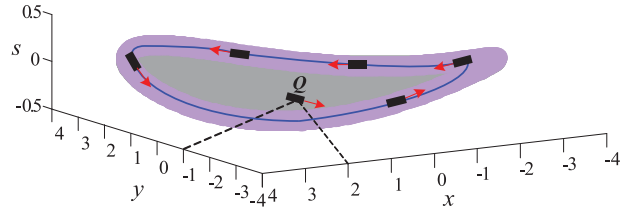


Figure 4. Reference circle trajectory.

Differentiation yields (Kanayama et al., 1990):

$$\dot{V}_2(\boldsymbol{e}_m, t) = -k_1 x_e^2 - k_3 \sin^2 \theta_e / k_2 \leq 0. \qquad (40)$$

This shows that the tracking position convergence is guaranteed by the control law (32). According to (38) and (40), we can obtain $\dot{V} = \dot{V}_1 + \dot{V}_2 \leq 0$, so that $V \to 0$ as $t \to \infty$. According to the standard Lyapunov theory (Slotine & Li, 1991), both the system stability and tracking convergence are guaranteed by the control law (32) driving the system (14), which closely tracks the desired motion trajectories.

5. Simulation results

The slip ratio of wheel is an important state variable. When slippage occurs between the wheels and the ground, the

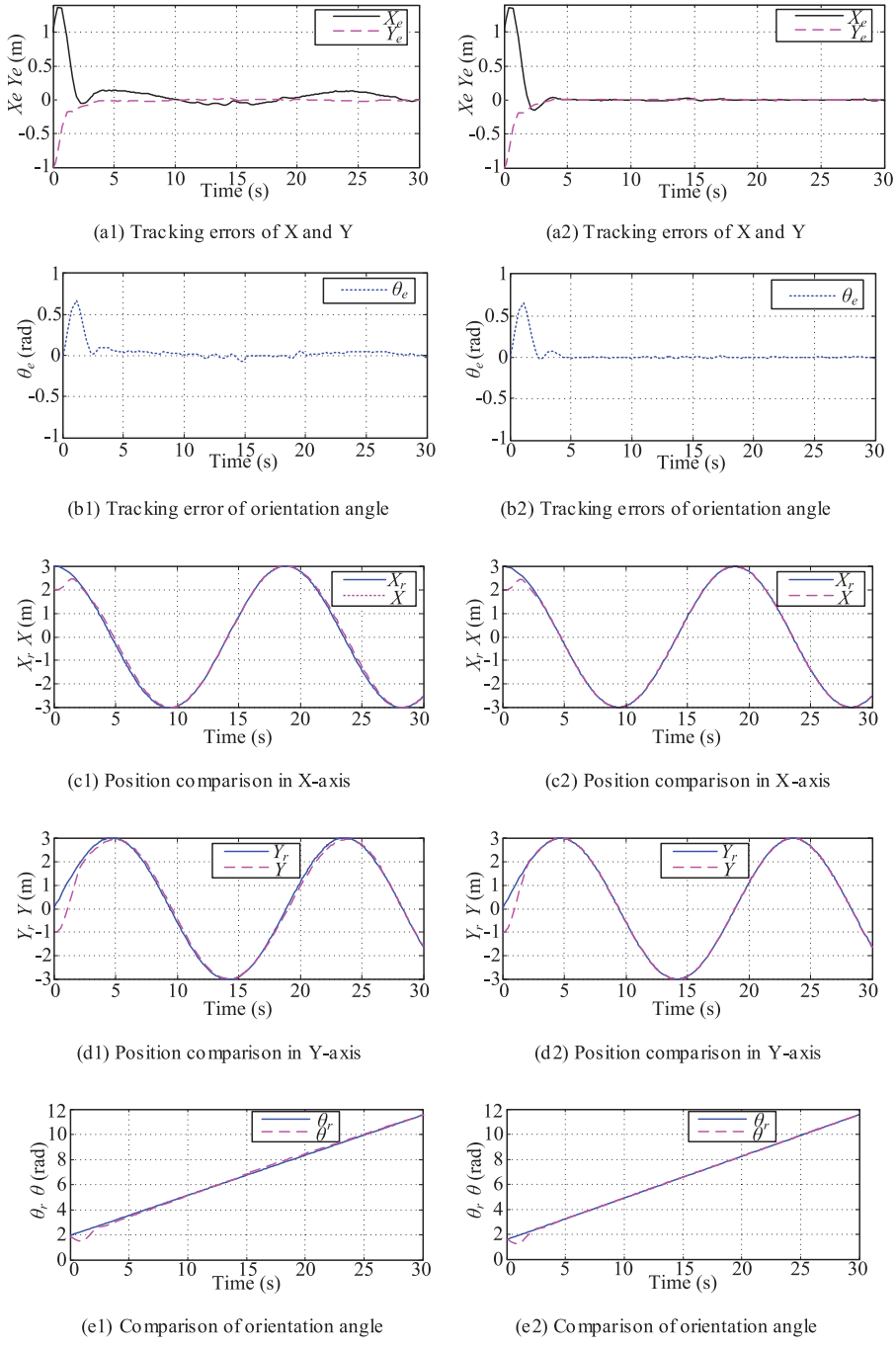


Figure 5. Artificial NNC without compensation (left). Adaptive NNC with compensation (right).

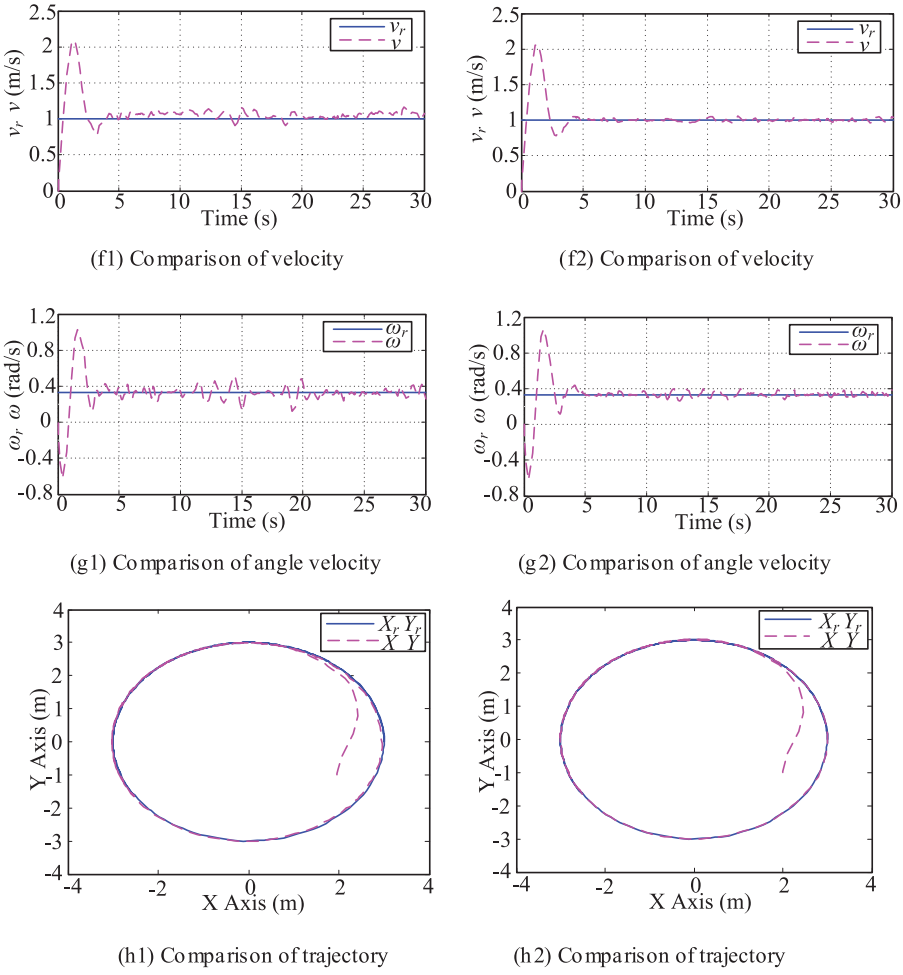


Figure 5. (Continued)

wheel velocities are influenced by the slippage, necessitating the analysis using an estimated slip ratio for movement control. According to the definition of the slip ratio (Ding et al., 2010), we know that the range of slip ratios is from -1 to 1, the change of slip ratios can resemble a sine curve, and the slip ratios will increase with increasing slope. Therefore, we can assume as follows.

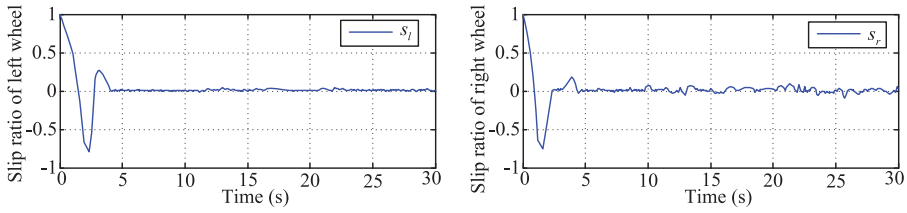
Let the desired trajectory be a space circular ring, $x_r = 3\cos(t/3)$, $y_r = 3\sin(t/3)$, $z_r = -0.4\cos(2t/3)$, and $t \in [0, 30]$s and where the initial position is $Q = (2, -1, -0.4)$ (Figure 4).

The slip ratio could be considered as the time variable function '0.2 sin (2t/3)', which increases or decreases with the slope angle of the reference trajectory.

We now demonstrate the adaptive NN control shown in Figure 3. Here, we compare its performance with slip-compensation and without slip-compensation when slip exits between the wheels and the ground. Two control performances were implemented and tested using Matlab Simulink models: A. Artificial NN controller (Jolly, Kumar, & Vijayakumar, 2009) under slippage; B. Adaptive NN controller with slip-compensation under slippage. We adopt vehicle parameters (Figure 1) as follows: $m = 5$ kg, $I = 4$ kgm^2, $L = 0.2$ m, $r = 0.15$ m, under the time varying external disturbance $\boldsymbol{\tau}_d = (\sin t, \cos t, 1)^{\mathrm{T}}$N, $v_r = 1$ m/s, $\omega_r = 1/3$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 behaviour: $k_1 = 2$, $k_2 = 9$, $k_3 = 6$, $k_4 = 20$, $k_5 = 10$. In the neural network, we selected a radial basis function with $N_h = 6$ hidden-layer neurons and $\kappa = 0.35$.

5.1 *Artificial NN control under slippage*

The response of artificial NN control (NNC) is described under the tangential slip of the wheels, as shown (a1-g1) in Figure 5. If we do not compensate the slippage or even neglect the effect of slippage in velocity control, these results of the tracking trajectory present large position and velocity error changes. Obviously, these results are not the desired precision for tracking control.



(i1) Real slip ratio of left wheel (i2) Real slip ratio of right wheel

Figure 6. Adaptive NNC with compensation. (i1) Real slip ratio of left wheel. (i2) Real slip ratio of right wheel.

5.2 *Adaptive NN control with slip-compensation under slippage*

The response of this controller is shown (a2–h2) in Figure 5. Lost velocity has been compensated due to wheel slippage. And it is obvious that both the posture and velocity tracking errors converge to the smaller values. Thus, when slippage occurred between the wheels and the ground, the control method with slip-compensation and adaptive NNC laws performed the better tracking effect during the motion control. Through this comparison, we can know that slip-compensation plays a significant role during the feedback of estimated slip ratios. Figure 6 (i1, i2) showed that the real slip ratios with slip-compensation is converging to the smaller scopes than the prior designed ones based on the terrain and trajectory.

6. Conclusion and future work

In this paper, an adaptive tracking control method has been proposed and demonstrated via neural networks and slip-compensation. This control scheme has been designed for tracking the desired motion trajectory in a WMR system under slip condition. In addition, we have developed a neural network learning procedure to enhance the performance of the proposed control scheme, even when tracking on uneven terrain. The stabilisation of the inner closed-loop system has been analysed. It has been shown that the convergence of the velocity and position tracking errors to zero is guaranteed by the proposed adaptive NNC law (32) based on slip-compensation. Even though there exist external disturbances or unknown system parameters, such as friction and slippage, which are very difficult to model using conventional techniques. Finally, it has demonstrated the results of precise tracking performance using a Matlab simulation.

This paper considers the compensation of calculating slip ratios, but not the interaction forces between wheels and ground. The future work is hoped to apply neural computing methods to terramechanics theory for terrain parameter estimation. The experiments need to be performed to compare with the system simulations under different trajectories and terrain.

Funding

This work was supported by the National Natural Science Foundation of China [grant number 50975059/61005080]; China Postdoctoral Science Foundation [grant number 20100480994]; Special Foundation [grant number 201104431]; '111' Project [grant number B07018].

References

Ding, L., Gao, H.B., Deng, Z.Q., Song, J., Liu, Y., Liu, G., & Lagnemma, K. (2011). Experimental study and analysis on driving wheels' performance for planetary exploration rovers moving in deformable soil. *Journal of Terramechanics, 48*, 27–45.

Ding, L., Gao, H.B., Deng, Z.Q., & Liu, Z. (2010). Slip-ratio-coordinated control of planetary exploration robots traversing over deformable rough terrain. *IEEE International Conference on Intelligent Robots and Systems*, 10, 4958–4963.

Endo, D., Okada, Y., Nagatani, K., & Yoshida, K. (2007). Path following control for tracked vehicles based on slip-compensating odometry. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 2871–2876). San Diego, CA.

Er, M.J., & Gao, Y. (2003). Robust adaptive control of robot manipulators using generalized fuzzy neural networks. *IEEE Transactions on Industrial Electronics, 50*(3), 116–129.

Fierro, R., & Lewis, F.L. (1998). Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks,* 9, 589–600.

Haykin, S. (2009). *Neural networks and learning machines* (3rd ed). Ontario: McMaster University Press.

Horn, R.A., & Johnson, C.R. (1985). *Matrix analysis*. New York: Cambridge Univ. Press.

Ge, S.S., Wang, J., Lee, T.H., & Zhou, G.Y. (2001). Adaptive robust stabilization of dynamic nonholonomic chained systems. *Journal of Robotic Systems, 18*, 119–133.

Iossaqui, J.G., Camino, J.F., & Zampieri, D.E. (2010). Adaptive tracking control of tracked mobile robots with unknown slip parameter. XVIII Congresso Brasileiro de Automática, 12 a 16-setembro, Bonito-MS, 1846–1851.

Jolly, K.G., Kumar, R.S., & Vijayakumar, R. (2009). An artificial neural network based dynamic controller for a robot in a multi-agent system. *Neurocomputing, 73*, 283–294.

Jung, J.R., & Yum, B.J. (2011). Artificial neural network based approach for dynamic parameter design. *Expert System with Applications, 38*, 504–510.

Kanayama, Y., Kimura, Y., Miyazaki, F., & Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. *Proc. IEEE International Conference on Robotics and Automation*, 1, 384–389