Neural Network-based Adaptive Tracking Control of Mobile Robots in the Presence of Wheel Slip and External Disturbance Force

Ngoc-Bach Hoang\(^1\), Hee-Jun Kang\(^2\)*

\(^1\)Graduate School of Electrical Engineering, University of Ulsan, Nam-gu, Ulsan, 680-749, South Korea
\(^2\)School of Electrical Engineering, University of Ulsan, Nam-gu, Ulsan, 680-749, South Korea

*Corresponding Author: E-mail: hjkang@ulsan.ac.kr, Tel: +82-52-259-2207, Fax: +82-52-259-1686

Abstract. In this paper, a novel adaptive tracking controller is proposed for mobile robots in presence of wheel slip and external disturbance force based on neural networks with online weight updating laws. The uncertainties due to the wheel slip and external force are compensated online by neural networks in order to achieve the desired tracking performance. The online weight updating laws are modified versions of the backpropagation with an e-modification term added for robustness. The global uniformly ultimately bounded stability of the system to an arbitrarily small neighborhood of the origin is proven using Lyapunov method. The validity of the proposed controller is confirmed by two simulation examples of tracking a straight line and a U-shape trajectory.

Keywords. Neural network, Adaptive control, Mobile robot dynamics, Wheel slip, Disturbance force

1. Introduction

In recent years, control of nonholonomic systems such as nonholonomic mobile robot has received wide attention and is a topic of great research interest due to the practical importance of its applications. The control researches for mobile robots have been centered on stabilization and tracking problems. Most proposed control algorithms have been proposed that rely on the kinematic model with nonholonomic assumptions such as backstepping [1], pure pursuit [2], neural networks [3], neural fuzzy [4], and linearization [5]. These algorithms completely neglect the robot dynamics. The control inputs, usually motor voltages, are assumed to instantaneously establish the desired robot velocities, known as perfect velocity tracking. In the case of heavy mobile robot with high speed, however, the above assumptions are not upheld in reality. In order to overcome this problem, some control schemes based on a full dynamic model have been proposed so far by taking into account the dynamic effects caused by mass, friction, and inertia [6]. Most of dynamic based controllers depend on the ideal of backstepping from kinematics into dynamics as proposed in [7]. Some control methods such as neural network [8, 9], fuzzy [10], robust damping [11], sliding model [12] have been proposed to enhance the robustness of tracking control performance. Despite having a better performance can be achieved, these algorithms mainly rely on the assumptions of rolling contacts without sliding motions, known as nonholonomic constraints. In most real applications of mobile robots, slip between the ground and wheel cannot be avoided due to various reasons such as uneven terrain, robot high speed, ground conditions, slippery surface. The tracking performance of the mobile robot could be seriously reduced because of the sliding motions. In order to
achieve the desired performance in the presence of slip, a controller which are robust to the slip between the wheel and the ground are significant.

The mobile robot freely moves in the ground based on the friction forces between the driving wheels and the ground, known as traction forces. Each traction force can be basically divided into two direction: longitudinal traction force and lateral traction force. The longitudinal traction force is proportional to the driving torque, which is usually supplied by electrical motor. Under the effect of that two forces, the wheel has ability to move forward or backward. In the acceleration or deceleration time, mobile robot always requires the high driving torque, which causes the high longitudinal traction force. If the longitudinal traction force exceeds the dynamic friction force, the sliding motion between the wheel and the ground in the longitudinal direction will occur, known as wheel slipping. The lateral traction force occurs under the effect of the centrifugal force generated when mobile robots tries to change its direction. If mobile robot with high speed suddenly changes its direction, the lateral traction force could not be enough to equalize the centrifugal force. Then the sliding motion in the lateral direction will occur, known as wheel skidding. In order to compensate only the effect of wheel slipping, an adaptive tracking controller is proposed and demonstrated via neural networks and wheel slip compensation in [13]. The slip ratios are assumed to be measurable using gyro-sensor and encoders. The approaches of using gyros and accelerometers to compensate for the slip in real time are also presented in [14, 15]. A robust controller taking care both slip-kinematic and slip-dynamic models is proposed in [16] using the framework of differential flatness. The simulation and experimental results in the slip-kinematic case show the effect of the proposed robust controller when the sliding motion occurs. However, in the case of the slip-dynamic, the robust controller requires the measurements of robot accelerations, which are very difficult to obtain in reality. In [17], the longitudinal traction force is included in an omni-directional mobile robot model by externally measuring the magnitude of slip. However, the ideal mobile robot model is used in control design for simplicity. In order to achieve position control, the authors in [18] have modeled the overall wheeled mobile robot as a third order under-actuated dynamic system with a second order nonholonomic constraint. The measurements of wheel slip are still assumed to be available for the controller design. The disadvantage of this assumption is due to the requirement of extra sensors such as gyroscope, accelerometer... to measure the wheel slip. In [19], a robust tracking controller is proposed in which the disturbance and state are estimated using a generalized extended state observer. The ideal of backstepping from kinematics into dynamics [7] is applied to find the required torque input. Another controller based on the estimate of the disturbance due to wheel slip can be found in [20]. In [21], the wheel slip and external loads are assumed to act as disturbances to the system. Then the sliding model control method [22, 23] is employed to design a tracking controller of the mobile robot. Even the boundary layer method is also applied, the chattering problem is needed to be considered more. And the bound of the uncertainties, which is pre-defined based on the knowledge about robot system, are required to design the tracking controller. Too large bounded value can cause too much chattering in control effort or reduce the tracking performance. The disadvantages of measurable wheel slip or uncertainty bounded motive us to develop a new adaptive controller in which those information are no longer needed.

Recently, neural networks are increasingly recognized as a powerful tool for controlling many complex dynamic systems thanks to the advantages such as learning ability, adaptation, flexibility [24-27]. In this
paper, we propose a new adaptive controller using an online tuning neural network to deal with the mobile robot uncertainties due to the wheel slip and external disturbance forces. First, the detail of dynamic model of mobile robot subject to wheel slip and external disturbance load is developed [21]. The friction forces are simply divided into two forces: lateral and longitudinal forces, which are generated due to the slip angle and the tire slip, respectively. The equation motion of the mobile robot can be obtained by the summation of the external forces and moments in the body centered reference frame based on Newton's Law. In order to reduce the harmful effect of the external loads and wheel slip on the control performance, an online learning neural networks, which does not require preliminary offline tuning, is implemented. The neural network weights are updated based on the backpropagation plus an e-modification term to guarantee its robustness. It is noted that the proposed controller does not require the assumption of uncertainty bounded or measurement of wheel slip. Finally, two simulation examples of tracking a straight line and a U-shape trajectory are performed to confirm the effectiveness of the proposed algorithm.

This paper is organized as follows: In Section 2, the kinematic and dynamic model of mobile robots in presence of wheel slips and external disturbance forces are described. In Section 3, the proposed adaptive tracking controller using online tuning neural networks is presented in details and the stability of the closed-loop system is proven. Two simulation examples of tracking a straight line and a U-shape trajectory are presented in Section 4. Finally, the research conclusions are given in Section 5.

2. Kinematic and Dynamic Model of Mobile Robot with Wheel Slip and External Disturbance Forces

In this section, we derive the kinematic and dynamic model of a differentially driven wheeled mobile robot shown in Fig.1. o-XY is the global coordinate system. G-xy is the coordinate system fixed to the mobile platform. G is the center of mass of the mobile robot. \( u \) and \( v \) are the longitudinal and lateral velocities at the center gravity of the mobile robot, respectively, and \( r \) denotes the yaw rate. \( \omega_r \) and \( \omega_l \) are the wheel angular velocities of right and left wheel, respectively.
The mobile robot is controlled through the velocities or torques of the driving wheels. The kinematic equations are derived based on the constraints of the velocity components. With the wheel velocities \( \omega_r \) and \( \omega_l \) are given, the longitudinal and lateral velocities \( u, v \) and the yaw rate \( r \) are given as follows

\[ u = \frac{1}{2}(u_r + u_l) \]  

\[ v = \frac{d}{2b}(u_l - u_r) + v' \]  

\[ r = \frac{l}{2b}(u_l - u_r) \]  

\[ u_r = R\omega_l - u_r' \]  

\[ u_l = R\omega_r - u_l' \]  

where \( u_r \) and \( u_l \) are the longitudinal speeds of the right and left wheel centers, respectively. \( v' \) is the wheel skidding. \( u_r' \) and \( u_l' \) are the wheel slipping of the right and left wheel centers, respectively. \( d \) is the distance from the wheel center to the geometrical center line of the platform. \( b \) is the distance from the wheel center line to the mass center of the platform. \( R \) is the wheel radius. The wheel slip model is considered by adding the three velocity components: \( v', u_r', u_l' \). The kinematic model in the case of no wheel slip can be obtained easily by setting the three above components to zeros.

The trajectory of the tracking point \( O \) and the heading angle of the mobile robot in the world coordinate \( o-XY \) are described as follows

\[ \dot{X} = u\cos \phi - (v+er)\sin \phi \]  

\[ \dot{Y} = u\sin \phi + (v+er)\cos \phi \]
Consider the mobile robot in the planar case with three degrees of freedom, the equation motion of the mobile robot can be obtained by the summation of the external forces and moments in the body centered reference frame based on Newton's Law as follows

\[
\dot{X} = \begin{bmatrix} u \cos \phi - (d + e) r \sin \phi \\ u \sin \phi + (d + e) r \cos \phi \\ r \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tau_y \\ \tau_r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

\[
\dot{Y} = \begin{bmatrix} mR^2 r^2 / \Omega \end{bmatrix}
\]

\[
\dot{\phi} = \begin{bmatrix} -d m R^2 v / \Omega \end{bmatrix}
\]

\[
\dot{r} = \begin{bmatrix} nR / \Omega \end{bmatrix}
\]

\[
\dot{\theta} = \begin{bmatrix} nbR / \Omega \end{bmatrix}
\]

where \( \Omega = m R^2 + 2 I_z; \Omega_r = 2 b^2 I_z + R^2 (I_z + m d^2); \tau_y = \tau_L + \tau_r; \tau_r = \tau_L - \tau_r \)

\[
\delta_u = \frac{1}{\Omega} \left( R^2 P_x + m R^2 v \dot{r} - I_z (\dot{u}_r + \dot{u}_s) \right); \delta_y = \frac{1}{\Omega} \left( P_y d R^2 - M_b R^2 - m d R^2 \dot{v}^2 - b I_z (\dot{u}_s - \dot{u}_r) \right)
\]

\[
\delta_\phi = -v \sin \phi; \delta_\theta = v \cos \phi
\]

The vector \( \delta = \begin{bmatrix} \delta_u & \delta_y & \delta_\phi & \delta_\theta \end{bmatrix}^T \) represents for the uncertainties due to the wheel slip and the external forces.
For the position tracking control, we take the world coordinates (Cartesian space) of tracking point O as the control variable from the posture vector as follows

\[
x = \begin{bmatrix} X \\ Y \end{bmatrix}^T \quad \text{and} \quad \dot{x} = \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix}^T
\]

(15)

By taking the time derivatives of the dynamic equation Eq.(14) and substituting the kinematic model, we obtain the dynamic equation in terms of the position vector in the world coordinate system as follow

\[
\ddot{x} = D(\phi)(\tau + S(\dot{x}, \phi, \delta_m) + \zeta(\ddot{x}, \phi, \delta_m)) \\
\dot{\phi} = r(\dot{x}, \phi, \delta_m)
\]

(16)

(17)

where

\[
D(\phi) = \begin{bmatrix}
\frac{nR}{\Omega_e} \cos \phi & -\frac{nRb(d+e)}{\Omega_e} \sin \phi \\
\frac{nR}{\Omega_e} \sin \phi & \frac{nRb(d+e)}{\Omega_e} \cos \phi
\end{bmatrix}
\]

\[
S = \begin{bmatrix}
-\frac{emR^2 + 2(d+e)I_e}{nR}r^2 \\
2b^2 I_e + R^2 (I_e - mde)
\end{bmatrix}
\]

\[
\zeta = \begin{bmatrix}
\frac{\Omega}{nR}(\delta_m - \dot{\nu}r) \\
\frac{\Omega}{nbR(d+e)}(b+e)(\delta_m - \dot{\nu}r)
\end{bmatrix}
\]

\[
r(\dot{x}, \phi, \delta_m) = \frac{I}{d+e} \left(-\left(\dot{X} - \delta_m\right)\sin \phi + \left(\dot{Y} - \delta_m\right)\cos \phi\right)
\]

Remark 1: The matrix \(D(\phi)\) is a positive definite matrix.

By setting all the slip conditions and the external forces to be zeros, the nonholonomic model of mobile robot can be described as follows

\[
\ddot{x} = D_0(\phi)(\tau + S_0(\dot{x}, \phi)) \\
\dot{\phi} = r(\dot{x}, \phi)
\]

(18)

(19)

where

\[
S_0 = \begin{bmatrix}
\frac{emR^2 + 2(d+e)I_e}{nR}r^2 \\
2b^2 I_e + R^2 (I_e - mde)
\end{bmatrix}
\]

\[
r(\phi, \dot{x}) = \frac{I}{d+e} \left(-\dot{X} \sin \phi + \dot{Y} \cos \phi\right)
\]
3. Neural Network-based Adaptive Tracking Control

3.1 Multi-layer neural networks

The capability of neural networks for identification, observation and control of nonlinear systems has been investigated in offline and online environment [9, 28, 29]. It is well known that a three-layer neural networks has capable of approximating nonlinear systems with some sufficiently large number of hidden layer neurons. In this sub-section, we briefly introduce about the three-layer neural network structure which is used in the next section. As shown in Fig. 2, the outputs of neural networks can be determined as follows

\[
y_i = \sum_{j=1}^{N_h} w_{ij} \sigma \left( \sum_{k=1}^{N_i} v_{jk} x_k + \theta_{ij} \right) + \theta_{wi}
\]  

where \( N_i, N_h, \) and \( N_o \) are the numbers of input-layer, hidden-layer, output-layer neurons, respectively. \( v_{jk} \) and \( w_{ij} \) are the adjustable weights of hidden-layer and output layer, respectively. The threshold offsets are denoted by \( \theta_{ij} \) and \( \theta_{wi} \). \( \sigma(\cdot) \) is the sigmoid activation function as

\[
\sigma(z) = \frac{2}{1 + e^{-2z}} - 1
\]

For convenience, Eq. (20) can be written in the following compact form

\[
y = W \sigma(V \bar{x})
\]  

where \( W \in \mathbb{R}^{N_o \times (N_h+1)} \), \( V \in \mathbb{R}^{N_i \times (N_h+1)} \) are the weight matrices. \( \bar{x} = [1, \bar{x}_1, \bar{x}_2, ..., \bar{x}_{N_h}]^T \in \mathbb{R}^{N_h+1} \)

\[
y = [y_1, y_2, ..., y_{N_o}]^T \in \mathbb{R}^{N_o}, \quad \sigma(V \bar{x}) = [1, \sigma(V_{e1} \bar{x}), \sigma(V_{e2} \bar{x}), ..., \sigma(V_{eN_o} \bar{x})]^T \in \mathbb{R}^{N_h+1}, \quad (V_e \text{ represents the } ith \text{ row detected from the matrix } V)
\]
matrix of $V$). It is emphasized that $\sigma(\cdot)$ is the map from $\mathbb{R}^{N_s}$ to $\mathbb{R}^{N_s+1}$. By this expression $\theta_{vi}$ and $\theta_{wi}$ are included as the first columns of $V$ and $W$, respectively. Therefore, any tuning of $V$ and $W$ will include tuning of the thresholds as well.

Let $\overline{X}$ be a compact simply connected set of $\mathbb{R}^{N_s}$ and $g(\overline{x})$ be a continuous function from $\overline{X}$ to $\mathbb{R}^{N_c}$.

Then, for any given positive constants $\varepsilon$, there exist ideal parameters $W^*, V^*, N_h$ such that

$$g(\overline{x}) = W^* \sigma(V^* \overline{x}) + \varepsilon$$

where $\varepsilon$ is the bounded neural network approximation error.

**Assumption 1:** The optimal weight matrices $W^*, V^*$ and the sigmoid activation function $\sigma(\cdot)$ are bounded such that $\|W^*\|_F \leq W_m, \|V^*\|_F \leq V_m, \|\sigma\|_F \leq \sigma_m$.

It is noted that the ideal parameters $W^*$ and $V^*$ are only quantities required for analytical purpose. The bounded values $W_m, V_m, \sigma_m$ are required for the stability analysis. The estimation of function $g(\overline{x})$ is given by

$$\hat{g}(\overline{x}) = \hat{W} \sigma(\hat{V} \overline{x})$$

The function approximation error is given by

$$\tilde{g}(\overline{x}) = g(\overline{x}) - \hat{g}(\overline{x}) = W^* \sigma(V^* \overline{x}) + \varepsilon - \hat{W} \sigma(\hat{V} \overline{x})$$

Using Taylor series expansion for $\sigma(V^* \overline{x})$, Eq.(24) can be rewritten as

$$\tilde{g}(\overline{x}) = \hat{W} \sigma(V^* \overline{x}) + \varepsilon_e$$

where $\varepsilon_e = \hat{W} \left( \sigma'(\hat{V} \overline{x}) \hat{V} \overline{x} + O^2(\hat{V} \overline{x}) \right) + \varepsilon$ with $\|\varepsilon_e\| \leq \varepsilon_m$ is a higher order approximation error which is very small in comparison with the first term in Eq.(26).

### 3.2 Adaptive tracking controller design

Let us denote

$$D(\phi) = D_0(\phi) + \Delta D(\phi)$$

$$S(\dot{x}, \phi, \delta_m) = S_o(\dot{x}, \phi) + \Delta S(\dot{x}, \phi, \delta_m)$$

where $D_0(\phi)$ and $S_o(\dot{x}, \phi)$ are the known functions as Eq.(18). $\Delta D(\phi)$ and $\Delta S(\dot{x}, \phi, \delta_m)$ are the uncertainty terms. Now, the dynamic model in Eqs.(16)(17) are rewritten in the following form

$$\ddot{x} = D_o(\phi)(\tau + S_o) + \Psi(\tau, \dot{x}, \phi, \delta_m)$$

where $\Psi(\tau, \dot{x}, \phi, \delta_m) = \Delta D(\tau + S + \zeta) + D_o(\Delta S + \zeta)$ is the vector of the robot unknown uncertainties due to the wheel slip and the external disturbance forces.

The approximation of the robot unknown uncertainties $\Psi(\tau, \dot{x}, \phi, \delta_m)$ can be written in the form of neural network Eq.(23) as follows

$$\Psi(\tau, \dot{x}, \phi, \delta_m) = W^* \sigma(V^* \overline{x}) + \varepsilon(\overline{x})$$
where $W^*$ and $V^*$ are the optimal weight matrices of the output and hidden layers, respectively. $\mathbf{x} = [\tau, \dot{x}, \phi]^T$ is the neural network input vector. $\varepsilon(\mathbf{x})$ is the bounded neural network approximation error.

The desired trajectory of mobile robot is given by $x_d, \dot{x}_d, \ddot{x}_d \in \mathbb{R}^2$, which is commonly defined by the concept of the virtual vehicle [8]. The tracking error is defined as follows

$$e = x - x_d$$
$$\dot{e} = \dot{x} - \dot{x}_d$$

The filtered tracking error is defined as

$$r = \dot{e} + \Lambda e = \dot{x} - \dot{x}_d$$

where $\Lambda = \Lambda^T > 0$ is the design parameter matrix. And $\dot{x}_r = \dot{x}_d - \Lambda e$ is defined as the reference velocity vector.

In this paper, the following adaptive tracking controller for the mobile robot in presence of the wheel slip and the external disturbance forces is proposed

$$\tau = D(\phi)(\dot{x}_r - Kr - \dot{\mathbf{W}} \sigma(\mathbf{V}) - S_0(\dot{x}, \phi))$$

where $K$ is a positive definite matrix, and $\dot{\mathbf{W}}(\tau, \dot{x}, \phi, \delta, \sigma) = \dot{\mathbf{W}} \sigma(\mathbf{V})$ is the output of neural network in order to compensate the effect of the wheel slip and the external disturbance forces. The block diagram of the proposed controller is shown in Fig. 3.

By inserting Eq.(34) into the dynamic model Eq.(29), we obtain the filtered tracking error as follows

$$\dot{r} = -Kr + W^* \sigma(V^\mathbf{x}) + \varepsilon(\mathbf{x}) - \dot{\mathbf{W}} \sigma(\mathbf{V})$$

**Fig. 3.** Block diagram of the neural network based adaptive tracking controller

Once the structure of the neural networks is known, a suitable adaptation tuning laws should be defined to train the network. In this paper, a modified version of the backpropagation with an e-modification term added for robustness is implemented in order for online weight updating. Lets define the objective function as $J = (1/2)r^T \tau$, where $r$ is the filtered tracking error Eq.(33).

The updating laws are obtained as follows [30]

$$\dot{\mathbf{W}} = -\alpha_1 \left( \frac{\partial J}{\partial \mathbf{W}} \right) - \beta_1 \| \dot{\mathbf{W}} \| \dot{\mathbf{W}}$$
$$\dot{V} = -\alpha_2 \left( \frac{\partial J}{\partial V} \right) - \beta_2 \| \dot{\mathbf{V}} \| \dot{\mathbf{V}}$$
where the first terms are the backpropagation terms and the second terms are the e-modification terms. \( \alpha_{1,2} > 0 \) are the learning rates, \( \beta_{1,2} > 0 \) are small positive numbers. The updating mechanism is the most important part in the adaptive controller in order to guarantee the stability of the closed-loop system. The numerous papers discussing adaptive control for the system in the presence of disturbances and uncertainties have been proposed in the last decades [35-38]. To improve the performance of the system while retaining the advantage of assuring the robustness in the presence of bounded disturbance, the adaptive mechanism are suitably modified using a dead zone [35] or a time-varying dead zone [36], \( \delta \)-modification [37], e-modification [38]. The e-modification is first proposed in [38] in which a term proportional to \( e \), where \( e \) is the output error, is added to the adaptive law. In this paper, the e-modification terms in the form \(-\beta_1 \|\dot{W}\|\dot{W}\) and \(-\beta_2 \|\dot{V}\|\dot{V}\) are employed to improve the performance of the adaptive updating neural network weights. Together with the backpropagation algorithm, the adaptive updating law (36)(37) is simple, possible for real time application. In addition, the system stability is easily proved by Lyapunov method in the Theorem 1.

Let us define

\[
\text{net}_W = \dot{W} \sigma(\dot{V}) \tag{38}
\]

\[
\text{net}_V = \dot{V} \sigma \tag{39}
\]

The backpropagation terms are computed as

\[
\frac{\partial J}{\partial W} = \frac{\partial J}{\partial r} \frac{\partial r}{\partial \text{net}_W} \frac{\partial \text{net}_W}{\partial W} \tag{40}
\]

\[
\frac{\partial J}{\partial V} = \frac{\partial J}{\partial r} \frac{\partial r}{\partial \text{net}_V} \frac{\partial \text{net}_V}{\partial V} \tag{41}
\]

From Eq.(35), we get

\[
\frac{\partial r}{\partial \text{net}_W} \approx -K^{-1} \tag{42}
\]

\[
\frac{\partial r}{\partial \text{net}_V} = -K^{-1} \dot{W}(I - \text{diag}(\sigma_i^2)) \tag{43}
\]

And from Eqs.(38)(39)

\[
\frac{\partial \text{net}_W}{\partial W} = \sigma(\dot{V}) \tag{44}
\]

\[
\frac{\partial \text{net}_V}{\partial V} = \sigma \tag{45}
\]

By inserting Eqs.(42)(43)(44)(45) into Eqs.(40)(41), the weight updating laws are obtained as

\[
\dot{W} = -\alpha_1 \left( r^T K^{-1} \right)^T \left( \sigma(\dot{V}) \right)^T - \beta_1 \|\dot{W}\|\dot{W} \tag{46}
\]

\[
\dot{V} = -\alpha_2 \left( r^T K^{-1} \dot{W}(I - \text{diag}(\sigma_i^2)) \right)^T \text{sgn}(\sigma) - \beta_2 \|\dot{V}\|\dot{V} \tag{47}
\]

In order to simplify the stability analysis, we replace \( \sigma \) in Eq.(45) by \( \text{sgn}(\sigma) \) as showing in Eq.(47). The weight adaptation error \( \ddot{W} = W^* - \dot{W} \) and \( \ddot{V} = V^* - \dot{V} \) can be written as
\[ \dot{W} = \alpha_1 \left( r^T K^{-1} \right)^T \left( \sigma(V) \right)^T + \beta_1 \| r \| \dot{W} \]  
\[ \dot{V} = \alpha_2 \left( r^T K^{-1} \dot{W} (I - \text{diag}(\sigma^2)) \right)^T \text{sgn}(\sigma^2) + \beta_2 \| r \| \dot{V} \]  

**Theorem 1**: Consider the mobile robot model in the presence of wheel slip and external disturbance force as shown in Eq. (29) with the proposed adaptive control Eq. (34). If the weight of neural networks are updated according to Eqs. (46)-(47). Then the filtered tracking error \( r \) is globally uniformly ultimately bounded. The weight adaptation errors \( \dot{W}, \dot{V} \) of neural networks are also bounded.

**Proof**: For analyzing the stability of the closed-loop system, the following Lyapunov positive definite function is considered

\[ L = \frac{1}{2} r^T r + \frac{1}{2} \text{trace}(W^T \dot{W}) + \frac{1}{2} \text{trace}(V^T \dot{V}) \]  

The time derivative of Eq. (50) is given by

\[ L = r^T \dot{r} + \text{tr}(W^T \dot{W}) + \text{tr}(V^T \dot{V}) \]  

Now, by substituting Eqs. (35)-(46) into Eq. (51), we get

\[ L = -r^T K r + r^T \left( W^* \sigma(V) \right) + (\sigma(V) - W^* \sigma(V)) \]  
\[ + \alpha_1 \text{tr}\left( W^T \left( r^T K^{-1} \right)^T \left( \sigma(V) \right)^T \right) + \beta_1 \text{tr}\left( W^T \| r \| (W^* - r) \right) \]  
\[ + \alpha_2 \text{tr}\left( V^T \left( r^T K^{-1} \dot{W} (I - \text{diag}(\sigma^2)) \right)^T \text{sgn}(\sigma^2) \right) + \beta_2 \text{tr}\left( V^T \| r \| (V^* - r) \right) \]  

Based on the properties of the matrix trace and sigmoidal function, we have the following inequalities

\[ \dot{W}^T (W^* - \dot{W}) \leq W_n \| \dot{W} - \dot{W} \| \]  
\[ \dot{V}^T (V^* - \dot{V}) \leq V_n \| \dot{V} - \dot{V} \| \]  
\[ \dot{W}^T \left( r^T K^{-1} \right)^T \left( \sigma(V) \right)^T \leq \sigma_n \| \dot{W} \| \]  
\[ \dot{V}^T \left( r^T K^{-1} \dot{W} (I - \text{diag}(\sigma^2)) \right)^T \text{sgn}(\sigma^2) \leq \| K^{-1} \| \| \dot{W} \| \| r \| (W_n + \dot{W}) \]  

where \( W_n, V_n \) are the upper bound of the neural network ideal weights \( W^*, V^* \) as \( \| W \| \leq W_n \), \( \| V \| \leq V_n \), and \( \sigma_n \) is the bounded of sigmoid function as \( \| \sigma(V) \| \leq \sigma_m \).

Then, inserting Eqs. (53)-(56) into Eq. (52), we have

\[ L \leq -\lambda_{\min}(K) \| r \|^2 + \left( \| \dot{W} \| + \| \dot{V} \| \right) + \alpha_1 \sigma_m \| K^{-1} \| \| \dot{W} \| \| r \| \]  
\[ + \beta_1 \left( W_n \| \dot{W} - \dot{W} \| \right) + \alpha_2 \sigma_n \| K^{-1} \| \| \dot{W} \| \| r \| + \beta_2 \left( V_n \| \dot{V} - \dot{V} \| \right) \]  
\[ \leq -\lambda_{\min}(K) \| r \|^2 + \left( \| \dot{W} \| + \| \dot{V} \| \right) \left[ \| \dot{W} \| + \alpha_1 \sigma_m \| K^{-1} \| + \beta_1 \| W_n \| \right] \]  
\[ + \left( \| \dot{V} \| + \alpha_2 \sigma_n \| K^{-1} \| + \beta_2 \| V_n \| \right) \]  

Now we assume that \( 2\beta_1 > \alpha_2 \| K^{-1} \| \) and \( 2\beta_2 > \alpha_2 \| K^{-1} \| \), and define the following auxiliary parameters
\[ A = \frac{\sigma_m + \alpha_1 \sigma_m \|K^{-1}\|}{2 \beta_1 - \alpha_2 \|K^{-1}\|} + \beta_1 W_n \]  
\[ B = \frac{\alpha_2 \|K^{-1}\| W_n + \beta_2 V_n}{2 \beta_2 - \alpha_2 \|K^{-1}\|} \]  

The Eq.(57) becomes

\[ \dot{L} \leq -\lambda_{\text{min}}(K) \|p\|^2 + \|r\|^2 - \left( e_n + \beta_1 - \frac{1}{2} \alpha_2 \|K^{-1}\| \right) \|W\|^2 + \beta_1 \|W\|^2 \]  
\[ \leq -\lambda_{\text{min}}(K) \|p\|^2 + \|r\|^2 - \left( e_n + \beta_1 - \frac{1}{2} \alpha_2 \|K^{-1}\| \right) \|W\|^2 + \beta_1 \|W\|^2 \]  

Therefore, the condition to guarantee the negative semi-definiteness of \( \dot{L} \) is as follows

\[ \|p\|^2 \geq \frac{e_n + \left( \beta_1 - \frac{1}{2} \alpha_2 \|K^{-1}\| \right) \|W\|^2 + \beta_1 \|W\|^2}{\lambda_{\text{min}}(K)} = r_0 \]  

This means the derivative of Lyapunov function \( \dot{L} \) is negative definite outside the ball with radius \( r_0 \), described by \( \Theta = \{ r \|p\| > r_0 \} \). The size of the ball radius \( r_0 \) can be kept small by proper selection of the learning rate \( \alpha, \beta > 0 \) and the damping factor \( \beta, \beta > 0 \). The above analysis shows the ultimately bounded of the filtered tracking error \( r \).

From Eqs.(46)(47), the weight errors can be rewritten as

\[ \hat{W} = -\beta_1 \|W\|^2 + \left( \alpha_1 \left( r^T K^{-1} \right)^T \left( \sigma(V) \right)^2 + \beta_1 \|W\|^2 \right) \]  
\[ \hat{V} = -\beta_2 \|V\|^2 + \left( \alpha_2 \left( r^T K^{-1} \hat{W} (I - \text{diag}(\sigma^2)) \right)^T \text{sgn}(\sigma) + \beta_2 \|V\|^2 \right) \]  

The second terms in Eqs.(62)(63) are bounded because \( r, \sigma, W, V, K, \hat{V} \) are bounded. Since, \( \beta, \beta > 0 \), the dynamic of Eqs.(62)(63) are also stable. Consequently, \( \hat{W}, \hat{V} \) are also bounded.

**Remark 2:** In order to estimate the robot dynamic parameters (mass, moment inertia) in Eq.(14), some offline identification methods [31, 32] such as the inverse dynamic identification method, the output error identification method are presented. In fact, it is very difficult to exactly obtain the dynamic parameters. The estimate results of the dynamic parameters are usually given with certain uncertainties, which can cause the divergence of the robot tracking error. Since, the proposed neural network-based adaptive tracking controller has ability to compensate the uncertainties due to wheel slip and external disturbance force, the uncertainties in dynamic model can be also compensated with the proposed controller.

**Remark 3:** The proposed controller requires the measurement of velocities in Cartesian space, which are difficult to obtain in real implementation. In order to overcome that drawback, the following super-twisting based velocity observer could be applied [33, 34]. By introducing the variables \( z_1 = x, z_2 = \dot{x} \), the model Eq.(29) can be rewritten in the state-space form
\[ \dot{z}_1 = z_2 \quad (64) \]
\[ \dot{z}_2 = D_0(\phi)(\tau + S_0) + \Psi(\tau, z_2, \phi, \delta_m) \quad (65) \]
\[ y = z_1 \quad (66) \]

The super-twisting observer [33] has the form
\[ \begin{align*}
\dot{\hat{z}}_1 &= \hat{z}_2 + \rho_1 \\
\dot{\hat{z}}_2 &= D_0(\phi)(\tau + S_0) + \rho_2
\end{align*} \quad (67) \]
\[ (68) \]
where \( \hat{z}_1 \) and \( \hat{z}_2 \) are the state estimations, and the correction variables \( \rho_1 \) and \( \rho_2 \) are the output injections of the form
\[ \begin{align*}
\rho_1 &= \lambda_1 [z_1 - \hat{z}_1]^{1/2} \text{sign}(z_1 - \hat{z}_1) \\
\rho_2 &= \lambda_2 \text{sign}(z_1 - \hat{z}_1)
\end{align*} \quad (69) \]
\[ (70) \]
Taking \( \hat{z}_1 = z_1 - \hat{z}_1 \) and \( \hat{z}_2 = z_2 - \hat{z}_2 \), we obtain the error equations
\[ \begin{align*}
\dot{\hat{z}}_1 &= \hat{z}_2 - \lambda_1 |z_1 - \hat{z}_1|^{1/2} \text{sign}(\hat{z}_1) \\
\dot{\hat{z}}_2 &= f(z_1, z_2, \hat{z}_1, \tau) - \lambda_2 \text{sign}(\hat{z}_1)
\end{align*} \quad (71) \]
\[ (72) \]
where \( f(z_1, z_2, \hat{z}_1, \tau) = D_0(z_1, z_2)(\tau + S_0(z_1, z_2)) - D_0(z_1, z_2)(\tau + S_0(z_1, z_2)) + \Psi(\tau, z_2, \phi, \delta_m) \). Suppose that the system states are assumed to be bounded, then the existence is ensured of a constant \( f_m \), such that \( |f(z_1, z_2, \hat{z}_1, \tau)| \leq f_m \) holds for any possible \( z_1, z_2 \).

With the super-twisting algorithm Eqs.(67-70), the states of the observer \( \hat{z}_1, \hat{z}_2 \) converge to the true states \( z_1, z_2 \) in finite time. We refer the reader to [33] for more details of the proof and how to select observer gains \( \lambda_1 \) and \( \lambda_2 \).

### 4. Simulation Results

To verify the performance of the proposed adaptive tracking controller in this paper, two simulation studies of tracking a linear and a U-shape trajectory have been carried out by Matlab/Simulink.

The robot parameter values used for simulations are shown in Table 1. The sampling time is set to \( 1 \text{(ms)} \).

The stable matrix \( K \) and \( \Lambda \) are set to \( K = \text{diag}[6, 6], \Lambda = \text{diag}[4, 4] \) for both simulations. In the neural network, the learning rate are \( \alpha_{1,2} = 30 \) and the damping factor are \( \beta_{1,2} = 3 \). These parameters are obtained by trial and error method. The initial value of the weight matrices are set to random numbers in \([0,1] \), \( \hat{W}_0 = \text{rand}(0,1), \hat{V}_0 = \text{rand}(0,1) \). The number of neurons in the hidden layer are 25 neurons.

<p>| ( m \text{(Kg)} ) | 40 | Mass of mobile robot |
| ( I_z \text{(Kg m}^2 ) | 4 | Moment of inertia of mobile robot |
| ( I_t \text{(Kg m}^2 ) | 0.1 | Moment of inertia of wheel |
| ( I_\phi \text{(Kg m}^2 ) | ( 1e^{-6} ) | Moment of inertia of rotor |
| ( R \text{(m)} ) | 0.16 | Wheel radius |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(b(m))</td>
<td>0.23</td>
<td>Half distance between the wheels</td>
</tr>
<tr>
<td>(d(m))</td>
<td>0.2</td>
<td>Position of robot mass center</td>
</tr>
<tr>
<td>(n)</td>
<td>100</td>
<td>Gear ratio</td>
</tr>
<tr>
<td>(e(m))</td>
<td>0.05</td>
<td>Position of tracking point</td>
</tr>
</tbody>
</table>

**Simulation Example 1**: Mobile robot is controlled to follow a straight line as shown in Fig. 4 (linear trajectory) with a trapezoidal velocity profile, which imposes a constant acceleration in the start phase, a cruise velocity, and a constant deceleration in the arrival phase. The desired trajectory are

\[
\begin{align*}
x_d & = \begin{cases} 1 + \frac{1}{2}t^2 & 0 \leq t \leq 2.5 \\ 1 + \frac{1}{12}(t-1.25)^2 & 2.5 < t \leq 7.5 \\ 5 - \frac{1}{4}(10-t)^2 & 7.5 < t \leq 10 \end{cases} (m) \\
y_d & = \begin{cases} 1 + \frac{1}{2}t^2 & 0 \leq t \leq 2.5 \\ 1 + \frac{1}{12}(t-1.25)^2 & 2.5 < t \leq 7.5 \\ 6 - \frac{1}{4}(10-t)^2 & 7.5 < t \leq 10 \end{cases} (m)
\end{align*}
\]  

(73)

The initial positions of the mobile robot are: \([x \ y \ \phi]^T = [0.9 \ 11 \ 0]^T\). The wheel skidding is assumed to be zero because the mobile robot follows a straight line. The wheel slipping are assumed to follow the sinusoidal function, respectively.

\[
v' = 0 \ (m/s) \tag{74}
\]

\[
u'_e = 0.2 + 0.2 \sin(3t) \ (m/s) \tag{75}
\]

\[
u'_i = 0.2 + 0.2 \sin(3t) \ (m/s) \tag{76}
\]

The external forces are set to \(P_x = 10 \ (N)\), \(P_y = 10 \ (N)\), \(M_x = 3 \ (N.m)\). We compare the controller Eq.(34) with the variable structure controller proposed in [21], in which the uncertainties due to the wheel slip and external forces are eliminated by the combination of the first order sliding mode and the boundary layer method as

\[
\tau = D_\phi \dot{\phi} (\dot{x}_d - \Lambda \ddot{x}_d - \Delta v - S_d \dot{x}_d \phi) \tag{77}
\]

where \(\Delta v\) is called modified variable structure control as

\[
\Delta v = \begin{cases} \frac{r}{\|x_m\|} x_m & \|x_m\| > \varepsilon \\ \frac{r}{\varepsilon} x_m & \|x_m\| \leq \varepsilon \end{cases}
\]

(78)

where \(0 < \varepsilon \leq 1\). In this paper, we select \(\varepsilon = 0.005\) and \(x_m = 2\). The tracking performance of the proposed controller is shown in Fig. 5. As shown in Fig. 6, both the proposed controller and the variable structure controller are able to track the desired trajectory in a finite time. However, as shown in Fig. 7.b, the variable structure causes much chattering in the torque control input even the boundary layer method is applied. There is no chattering with the proposed controller as shown in Fig. 7.a. Fig. 8.a and 8.b show the results of tuning the output weights and hidden weight of the neural network. We can see that all these parameters are guaranteed to be bounded and the online tuning algorithms Eqs.(53)(54) are converged.
Fig. 4. The linear trajectory of tracking control

Fig. 5. Linear trajectory tracking performance: (a) Position, (b) Velocity
Fig. 6. Linear trajectory tracking errors: (a) Proposed controller (b) Variable structure controller

Fig. 7. Torque control inputs in the case of linear trajectory tracking
(a) Proposed controller  (b) Variable structure controller
Simulation Example 2: In order to show the ability of the proposed algorithm to adaptively compensate the effect of wheel lip and disturbance forces, the mobile robot is controlled to track a U-shape trajectory as shown in Fig. 9 with a sudden wheel sliding motion. The desired trajectory are divided into three stage: In first stage, mobile robot is accelerated to reach a certain velocity in a straight trajectory. In this stage, it is assumed that there exists only the wheel slipping. In second stage, mobile robot is turned to follow a circle trajectory. Both the wheel slipping and wheel skidding are assumed to occur in this stage. Finally, mobile robot is decelerated to stop in a straight trajectory. In this final stage, only the wheel slipping is assumed to occur. The desired trajectory is described by the following function

\[
\begin{align*}
x(t) &= \begin{cases} 
1 + \frac{v_0^2}{2a}t^2 & 0 \leq t \leq 5 \\
1 + \frac{v_0^2}{2a} + 2.5 \cos\left(\frac{2\pi}{5}(t-5) - \frac{\pi}{2}\right) & 5 < t \leq 15 \\
1 + \frac{v_0^2}{2a}(20-t)^2 & 15 < t \leq 20
\end{cases} \\
y(t) &= \begin{cases} 
1 & 0 \leq t \leq 5 \\
3.5 + 2.5 \sin\left(\frac{\pi}{5}(t-5) - \frac{\pi}{2}\right) & 5 < t \leq 15 \\
6 & 15 < t \leq 20
\end{cases}
\end{align*}
\]  
(79)

The initial positions of the mobile robot are: \( [x \ y \ \dot{\phi}]^T = [0.9 \ \ \sqrt{2}] \). The wheel skidding and wheel slipping are assumed as follows, respectively

\[
\nu^r = \begin{cases} 
0 & 0 \leq t \leq 5 \\
0.3 + 0.1 \cos\left(\frac{\pi}{4}t\right) & 5 < t \leq 15 \ (\text{m/s}) \\
0 & 15 < t \leq 20
\end{cases}
\]  
(80)

\[
u^l = 0.2 + 0.2 \sin(3t) \ (\text{m/s})
\]  
(81)

\[
u^s = 0.2 + 0.2 \sin(3t) \ (\text{m/s})
\]  
(82)
The external forces are set to $P_x = 60 \, (N)$, $P_y = 60 \, (N)$, $M_x = 5 \, (N.m)$. In this simulation, the super-twisting observer Eqs. (64-67) is also applied to estimate the robot velocities in Cartesian space. The observer gains are selected as $\lambda_1 = 3$ and $\lambda_2 = 6$. The result of velocity estimation is shown in Fig. 10. The estimate velocities are confirmed to converge to the true values in finite time. The tracking performance and tracking errors in the case of U-shape tracking control are shown in Fig. 11 and Fig. 12. As shown in Fig. 12, the effect of the wheel skidding at $t = 5$ and $t = 15$ are also compensated by neural networks to obtain good tracking performance. Fig. 13 shows the torque control input of the proposed controller. Fig. 14.a and 14.b show the results of tuning the output weights and hidden weight of the neural network. It can be seen that the convergence of the online tuning algorithms can also be achieved. Fig. 15 show the capability of neural network in tracking the system uncertainties. Due to the highly nonlinear of neural network, the neural network weights can be changed to another local minimum in the training process. It causes the weights slightly change as shown in Fig. 14. However, this problem does not affect the tracking capability of neural network, which can be confirmed in Fig. 15.

Fig. 9. The U-shape trajectory of tracking control
Fig. 10. Super-twisting based velocity observer

Fig. 11. U-shape trajectory tracking performance: (a) Position, (b) Velocity
Fig. 12. U-shape trajectory tracking errors

Fig. 13. Torque control input in the case of U-shape trajectory tracking
5. Conclusions

In this paper, a novel adaptive tracking control of the mobile robot subject to the wheel slip and the external disturbance forces have been presented. The effect of wheel slip and external forces are considered as the uncertainties of the robot dynamic and are compensated by neural networks with an online weight updating laws. The weight updating mechanism is based on the modified backpropagation algorithm plus an e-modification term to guarantee the neural network robustness. The stability of the overall system is proven by Lyapunov direct method. The tracking error can be reduced as small as
desired according to the theoretical analysis. In addition, the super-twisting-based velocity observer is also presented to overcome the disadvantage of the proposed controller in Cartesian space. Detailed simulation results of two simulation examples have been given to confirm the effectiveness of our proposed algorithm.

Acknowledgments. This work was supported by Special Research Fund of Electrical Engineering in University of Ulsan

References


