Robust backstepping and neural network control of a low-quality nonholonomic mobile robot

Qiuju Zhang\textsuperscript{a}, James Shippen\textsuperscript{b,\*}, Barrie Jones\textsuperscript{c}

\textsuperscript{a}School of Manufacturing Engineering, Nanjing University of Science and Technology, Nanjing 210014, P.R. China
\textsuperscript{b}School of Manufacturing and Mechanical Engineering, The University of Birmingham, Birmingham B15 2TT, UK
\textsuperscript{c}School of Engineering, The University of Aston, Birmingham B4 2A, UK

Received 26 August 1998

Abstract

A robust motion controller based on neural network and backstepping technique is proposed for a two-DOF low-quality mobile robot (MR). There are two main components in the motion controller. One is the tracking controller, which guarantees the MR follows the reference trajectory; the other one is the wheel-level inverse NN controller, which compensates the dynamics of the MR. Simulation results are provided to validate the proposed controllers. Experiments with a real low-quality MR, which were built from cheap drivelines, have been used to verify the effectiveness and robustness of the motion controller. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Neural network control; Nonholonomic mobile robot

1. Introduction

In the last decade there has been enormous activity in the study of nonholonomic mechanical systems. The characteristic of the nonholonomic system is that the constraints, which are imposed on the motion, are not integratable, i.e., the constraints cannot be written as time derivatives of some functions of the generalised co-ordinates. The typical examples of nonholonomic control systems are: mobile robots, robot manipulators, wheeled vehicles and space robotics. Research continues on models of nonholonomic control systems, and on control design for motion planning and stabilisation [1,6]. Modelling and controlling such nonholonomic systems is a nontrivial prob-
lem. Even in the simplest case, a two-DOF mobile robot which we shall study here, the tracking control and pose (position and orientation) stabilisation require a sophisticated controller [1–3,6]. The relative difficulty of the control problem depends not only on the nonholonomic nature of the system but also on the control objective. Furthermore, any efforts at controlling the nonholonomic system with non-linear dynamics, uncertainties and disturbances will lead to more complex-structured controllers. For the tracking control problem of the MR, using smooth static time invariant state feedback for a velocity-controlled MR with one nonholonomic constraint was developed by Kanayama et al. [8] and later improved by Oelen and van Amerongen [7] with the effect that the performance of the controller depends only on the geometry of the reference trajectory rather than the reference linear velocity. Applications of the backstepping technique to the adaptive and robust control of the nonholonomic system were considered in Jiang and Nijmeijer [3], Kolmanovsky and McClamroch [4], and Wan and Lewis [5]. The design technique to obtain a suitable time-varying state feedback was based on an integrator backstepping procedure. Fierro and Lewis [2] also used the backstepping control approach to take into account the specific dynamics to convert a steering system command into control inputs for the actual vehicle. Neural network and fuzzy logic control approaches were used to deal with the disturbances and dynamic uncertainties in the MR [2,5,9].

Although significant progress has been made, some important research problems remain unsolved, such as the problem of robustness and control of nonholonomic systems when there are model uncertainties, as arises from parameter variations or from neglected dynamics, and model perturbations that destroy the “nonholonomic assumption”, e.g., the no-slip condition may only hold approximately. General methods for the design of robust controllers for nonholonomic systems are still unavailable [1]. Most researchers demonstrate their theoretical approaches by computer stimulations. However, simulation studies usually neglect practically important aspects, such as non-linear dynamics, rolling friction, compliance of the mechanical structure and unmodelled disturbances, therefore, the value of these approaches is limited.

Most mobile robots built for research use expensive precision motors to drive the wheels. If the MR is to become an inexpensive mass-produced item, then it must use inexpensive components. The control of a low-quality MR is more difficult than that of a high-quality one because the non-linearities and uncertainties of the MR are more significant and the measured feedback signals are rather noisy. Reports on the study of the control of low-quality mobile robots are unknown, but it is (from an engineering perspective) a very interesting problem and may provide potential practical applications.

The purpose of this paper is to use the backstepping technique and neural networks in the tracking problem for the two-DOF mobile robot. In particular, under our proposed controllers, a low-quality experimental MR, which was constructed from cheap drivelines, can follow a reference trajectory such as straight lines and circles. Special attention is paid to the controller implementations. The motion controller is comprised of two main components: one is the tracking controller, which is based on the backstepping technique and designed for a system with velocity input. Another one is the wheel-level controller, which controls the motion of the driven wheels. Two inverse dynamic controllers based on the neural networks were developed to control the velocity of the wheels. Between these two components, there were the decoupling and coupling interfaces which were used to decompose the linear velocity and angular velocity of the MR into the rotation speed of the two driven motors, and vice versa to synthesise the individual measurement signals from each wheel into the position and velocity of the MR.
The remainder of the article is organised as follows: Section 2 provides the theoretical background of a nonholonomic MR and tracking problem formulation. In Section 3, the tracking model and tracking control structure are presented. The simulation results of the proposed tracking controller are given in Section 4. Section 5 describes the experimental low-quality MR system and control implementation. In Section 6, the design of the inverse neural network controller is explained. In Section 7, experimental results are given. Finally, conclusions are presented in Section 8.

2. Problem formulation

2.1. Preliminary definitions

Consider a two-DOF mobile robot, which contains two driven wheels and a castor to carry the mechanical structure (see Fig. 1). It is a typical example of a nonholonomic mechanical system.

Two bases are used here to specify the position and orientation of the MR: an inertial Cartesian frame \(\{O, X, Y\}\) linked to the world and \(\{C, S, T\}\) linked to the mobile platform. It is assumed that the centre of mass of the MR is located in \(C\). The pose of the MR is completely specified by the vector \(p = [x_c, y_c, \theta]^{T}\) where \(x_c, y_c\) are the co-ordinates of \(C\) in the basis \(\{O, X, Y\}\), and \(\theta\) is the orientation of the basis \(\{C, S, T\}\) measured from the \(X\)-axis. The MR can be regarded as a system with two inputs: linear velocity \((v)\) and angular velocity \((w)\), both relative to the world; \(v = [v \ w]^{T}\). In a real MR system, \(x_c, y_c, \theta, v, w\) can all be measured or estimated.

The nonholonomic constraint assumes that the MR satisfies the conditions of pure rolling and non-slipping, that is, the MR can only move in the direction normal to the axis of the driving wheels [1–3]:

\[
y_c \cos \theta - x_c \sin \theta - d\dot{\theta} = 0
\]

(1)

![Fig. 1. A two-DOF mobile platform.](image)
2.2. Models of a nonholonomic mobile robot

Generally, a nonholonomic mobile robot system with \( n \) generalised configuration variables \( (q_1, \ldots, q_n) \) and subject to \( m \) constraints can be described by Eqs. (1), (2) and (6):

\[
M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(\dot{q}) + \tau_d = B(q)\tau - A^T(q)\lambda
\]

where \( M(q) \in \mathbb{R}^{n \times n} \) is a symmetric, positive definite inertia matrix, \( V_m(q, \dot{q}) \in \mathbb{R}^{n \times n} \) is the centripetal and coriolis matrix, \( F(\dot{q}) \in \mathbb{R}^{n \times 1} \) denotes the surface friction, \( \tau_d \) denotes bounded unknown disturbances including unstructured unmodeled dynamics, \( B(q) \in \mathbb{R}^{n \times r} \) is the input transformation matrix, \( \tau \in \mathbb{R}^{n \times 1} \) is the input vector, \( A(q) \in \mathbb{R}^{m \times n} \) is the matrix associated with the constraints, and \( \lambda \in \mathbb{R}^{m \times 1} \) is the vector of constraint forces.

It is difficult to use Eq. (2) for control purposes directly, as they require that the dynamics of the MR be completely known. In fact, perfect knowledge of the MR parameters is unattainable. Also unknown disturbances, friction \( F(\dot{q}) \) is very difficult to model by conventional techniques. So the kinematic model of an ideal MR is widely used in the MR control [1–3]. For a two-DOF mobile robot, the kinematic model can be given as:

\[
\begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & -d \sin \theta \\
\sin \theta & d \cos \theta \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
v \\
w
\end{bmatrix}
\]

From the control perspective, the motion control task of a real MR can be divided into two stages. The first is the tracking control, which generates the desired velocity profiles for the MR to follow a reference trajectory. The second stage is the wheel-level control, which controls the motion of the drive-wheels. Here we consider the design problems of both stages. In order to simplify the problem formulation, it is assumed that \( d = 0 \). The alternative formulations can be readily deduced when \( d \neq 0 \).

2.3. Tracking problem formulation

Suppose the MR is required to follow a reference trajectory (or a reference MR), with position \( p_r = [x_r \ y_r \ \theta_r]^T \) and velocity \( v_r = [v_r \ w_r]^T \).

The position error can be presented by

\[
e_p = p_r - p = [e_x \ e_y \ e_\theta]^T
\]

By using the transform matrix

\[
T_c = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]
The position error can be made orientation-independent:
\[ \mathbf{e}_p^c = \mathbf{T} \mathbf{e}_p = [e_s, e_t, e_o]^T \] (6)

or
\[
\begin{bmatrix}
    e_s \\
    e_t \\
    e_o
\end{bmatrix} =
\begin{bmatrix}
    e_s \cos \theta + e_y \sin \theta \\
    -e_s \sin \theta + e_y \cos \theta \\
    e_o
\end{bmatrix}
\] (7)

\( \mathbf{e}_p^c \) is measured using the MR basis \{C, S, T\} (see Fig. 1). \( e_s \) denotes the error in the driving direction, \( e_t \) denotes the lateral error and \( e_o \) is the orientation error.

The derivative of the position error is:
\[
\begin{bmatrix}
    \dot{e}_s \\
    \dot{e}_t \\
    \dot{e}_o
\end{bmatrix} =
\begin{bmatrix}
    w e_t - v + v_r \cos e_o \\
    -w e_x + v_r \sin e_o \\
    w_r - w
\end{bmatrix}
\] (8)

Therefore, it is necessary to find appropriate velocity control law \( \mathbf{v}_d = [v_d, w_d]^T \) of the form
\[ \mathbf{v}_d = [f_s(e_s, e_t, e_o, v_r, w_r), f_w(e_s, e_t, e_o, v_r, w_r)]^T, \]
such that \( \mathbf{p}_r \rightarrow \mathbf{p} \) as \( t \rightarrow \infty \). Then with the wheel-level controllers, \( \mathbf{v} \rightarrow \mathbf{v}_d \) as \( t \rightarrow \infty \).

In the following section, a tracking model for computing the desired \( \mathbf{v}_d \) will be examined.

3. Tracking model and controller structure

Among the three components of the position error, \( e_o \) has a significant effect on the position accuracy of the MR as it can result in the lateral error \( e_t \) directly. In the tracking error model (8), \( e_t \) is not directly controlled due to the nonholonomic constraint of the MR. To overcome this difficulty, the integrator backstepping method has been used by many researchers [2–5].

The possible conditions of \( e_t \) and \( e_o \) are given in Fig. 2. Three of them (denoted in a, b and c in bold lines) are considered as the best ones. a is the ideal situation when \( e_t = 0 \) and \( e_o = 0 \). If one wants to steer the MR towards the reference trajectory while the lateral error \( e_t \) is non-zero, b and c are the possible choices, which means it is better to have an orientation error to steer \( e_t \) to zero. Therefore, a new variable \( \bar{e}_o \) is introduced as \( \bar{e}_o = e_o + a e_t \), where \( a \) is a positive constant.

Suppose \( e_s \) and \( \bar{e}_o \) converge to zero, the following equation can be derived from Eq. (8):
\[ \dot{e}_t = -v_r \sin(\alpha e_t), \]
which is stable at \( e_t = 0 \). It can be proved that if \( e_s \rightarrow 0 \) and \( \bar{e}_o \rightarrow 0 \), then \( e_t \) shows geometric convergence [7].

Eq. (8) can be transformed into a new error model:
\[
\begin{bmatrix}
    \dot{e}_s \\
    \dot{e}_t \\
    \dot{e}_o
\end{bmatrix} =
\begin{bmatrix}
    we_t - v + v_r \cos e_o \\
    -we_x + v_r \sin e_o \\
    w_r - (1 + \alpha e_t)w + \alpha v_r \sin e_o
\end{bmatrix}
\] (9)
The backstepping technique [3,4] is used here to construct a feedback control law for Eq. (9). Consider the candidate Lyapunov function

$$V(t, e_s, e_t, e_{\theta}) = \frac{1}{2} e_s^2 + \frac{1}{2} e_t^2 + \frac{1}{2\gamma} \dot{e}_{\theta}^2$$

(10)

with $\gamma > 0$. From Eqs. (9) and (10), the time derivative of $V$ is:

$$\dot{V}(t, e_s, e_t, e_{\theta}) = e_s(-v + v_t \cos e_{\theta}) + e_t v_t \sin e_{\theta} + \frac{1}{\gamma} \dot{e}_{\theta} - w_t - (1 + \alpha e_{\theta})w$$

(11)

A smooth time-periodic feedback law can be chosen as [4]:

$$\begin{bmatrix} v_d \\ w_{\text{d}} \end{bmatrix} = \begin{bmatrix} k_1 e_s + v_t \cos e_{\theta} \\ (1 + \alpha e_{\theta})^{-1}(w_t + k_2 \dot{e}_{\theta} + \alpha v_t \sin e_{\theta}) \end{bmatrix}$$

(12)

where, $k_1$, $k_2$, and $\alpha$ are positive constants.

Eq. (12) is the same as the local tracking controller proposed by Jiang and Nijmeijer [3] replacing $\varphi(v_t e_t)$ by $\alpha e_{\theta}$.

However, the control law (12) is that $w$ may not be defined for every $t$; consider the case when
\(\alpha = 1\) and \(e_s = -1\). So a modification of Eq. (12) is required resulting in the tracking control law implemented:

\[
\begin{bmatrix}
    v_d \\
    w_{d,j}
\end{bmatrix} =
\begin{bmatrix}
    k_1 e_s + v_r \cos \theta \\
    w_r + k_2 \theta + \alpha v_r \sin \theta
\end{bmatrix}
\]

(13)

Note that Eq. (13) is similar to the control laws used by Fierro and Lewis [2]:

\[
\begin{bmatrix}
    v_d \\
    w_{d,j}
\end{bmatrix} =
\begin{bmatrix}
    c_1 e_s + v_r \cos \theta \\
    w_r + c_2 \theta e_r + c_3 v_r \sin \theta
\end{bmatrix}
\]

(14)

where, \(c_1, c_2, c_3\) are positive constants.

The simulation results, which will be provided later, show that the controller (13) has a better transient response than controller (14), and it is also more robust.

Fig. 3 gives the controller structure for a two-DOF MR, where \(p_r\) and \(v_r\) denotes the reference position and velocity vector respectively, \(v_d\) denotes the input vector for the system. \(q\) represents the measured displacements of the MR servos. The controller is comprised of two components: one is the tracking controller, which uses the backstepping technique to calculate the feedback law (13), and the other is the wheel-level inverse controller, which uses a neural network to compensate for the dynamics of the MR.

The coupling interface was used to decompose \(v_d\) into the command speeds of the two driven motors, that is,

\[
\begin{bmatrix}
    w_{m1} \\
    w_{m2}
\end{bmatrix} =
\begin{bmatrix}
    G_1/r_1(v_d + b w_d) \\
    G_2/r_2(v_d - b w_d)
\end{bmatrix}
\]

(15)

where \(G_1, G_2\) are the gear ratios of the two drivelines, \(r_1\) and \(r_2\) are the radii of the two driven wheels, respectively.

The coupling interface was used to compose the measured signals from encoders to the position measurements of the MR. In the present situation, it was done by integrating wheel displacements (odometry).

![Fig. 3. Structure of motion controller for the MR.](image-url)
4. Simulation results

The model for the simulations is shown in Fig. 4, together with the linear and angular velocity perturbations $\xi_v$ and $\xi_w$. For the real MR, the perturbations are primarily caused by mechanical disturbances, such as stick-and-slip, gear backlash, misalignment of the drive wheels, vibrations of the motors, noisy measurement signals, etc. All simulations were carried out using MATLAB.

Firstly the comparison of control law (13) and (14) was made by two sets of simulation results.

1. Ideal MR: $\xi_v = 0, \xi_w = 0$. The controller parameters for (13) were chosen by try-and-trial for best response: $k_1 = 10, k_2 = 2, \alpha = 4$; the parameters for controller (16) were the same as in [2]: $c_1 = 10, c_2 = 5, c_3 = 4$. The reference trajectory was a line with orientation angle $\theta = 45^\circ$; $v_r$ is shown in Fig. 5, $C_v = 2$ m/s; $w_r \equiv 0$. The initial errors were: $e_x = -0.6, e_y = 0, e_\theta = -\pi/4$.

2. As (1), but the reference trajectory is a circle with radius $v_r/w_r$; $v_r \equiv 1, w_r \equiv 1, x_r(t) = 1 + \sin(w_r t), y_r(t) = 1 - \cos(w_r t)$. Initial errors: $e_x = 1, e_y = 0.6, e_\theta = 0$. The simulation results are given in Figs. 6 and 7. It shows that controller (13) has a faster convergence speed and smaller position errors than controller (14). In order to illustrate the robustness of the tracking controller (15), another three sets of simulations were considered:

3. As (1), but with different initial errors: (a) initial error $(e_x, e_y, e_\theta) = (-0.06, 0, -\pi/4)$; (b) initial error $(e_x, e_y, e_\theta) = (-10, 2, -\pi/4)$; (c) initial error $(e_x, e_y, e_\theta) = (0, -5, \pi/2)$.

4. As (1), but with different $v_r$, that is, different $C_v$: (a) $C_v = 2$; (b) $C_v = 3$; (c) $C_v = 1$.

5. Controller parameters: same as (1). Reference trajectory: a square comprised of line and circle (see Fig. 10(a)); reference velocity $v_r$ and $w_r$ (see Fig. 10(b)); initial tracking error: $(e_x, e_y, e_\theta) = (0.1, 0.1, 0)$. Different signals were chosen as the perturbations: white noise, Coulombic friction and deadband as may exist in the control of a real MR.

Fig. 5. Reference linear velocity profile (simulation).
The simulation results are given in Figs. 8–10. Fig. 8 shows that for small (a) and large (b, c) initial errors, the tracking controller performs well. With large initial orientation error $e_{\theta}$, the MR exhibits oscillations whilst tracking the reference trajectory. Therefore, the lateral error decreases quickly while the orientation error remains. Fig. 9 shows the influence of different linear velocities on the tracking performance. For lower $C_v$, the convergence of the lateral error $e_t$ becomes slower and exhibits smaller overshoot. Fig. 10(a) gives the reference square and the actual path of the MR. The MR follows the reference square accurately except in the beginning and the middle arc part. Fig. 10(b) gives the control linear and angular velocities in different situations. It shows that the influence of perturbations on the linear velocity is more significant than on the angular velocity. Fig. 10(c) shows that the white noise causes noise-like errors, and Coulumbic friction and deadband generate steady-state errors in driving direction, but no lateral and orientation errors are introduced.

It is known that the most important performance criteria for tracking control of the MR are lateral and orientation errors. The simulation results in Figs. 8–10 show that the proposed tracking controller behaves well and has a good robustness.
5. A low quality experimental MR system

The experimental mobile robot system used in this paper is shown in Fig. 11. It consists of a vehicle with two driving front wheels mounted on the same axis; a rear castor wheel prevents the robot from tipping over as it moves on a plane. The motion and orientation are achieved by independent actuators, e.g., each front wheel is driven by a dc motor. The dc motor and driveline were built from low quality materials. The motor is a three-pole permanent magnetic dc motor. The driveline consists of a set of low-precision polythene worm-pinion gears. The gear ratio is 54:1. Both motor and driveline systems suffered from a lot of disturbances and uncertainties caused by the vibration of the motor, gear backlash and stick-and-slip problem. The displacement output of each motor was measured by a crude incremental encoder on the motor shaft (four-line encoder). No pre-filtering was used to filter the noisy feedback signals. Velocity signals of the servo were obtained by differentiating the displacement outputs. The error bound of the displacement output of the wheel was ± 0.03 rad (2π*gear ratio/4) and the velocity output was ± 0.3 rad/s with a sampling period of 0.1 s. The tracking controller was implemented in a PC 486-33. A 12-bit resolution D/A card and a 16-bit encoder card were used to transfer the control and feedback signals.

Fig. 12 gives the average deviations of the left and right motors’ velocities against input voltages on the motors from 10 trials. The input voltages varied from 1 to 5 V in steps of 0.1 V. The steady velocities of the motors were measured over a period of 5 s with a sampling period of 0.1 s. It shows that the velocity outputs were exceptionally noisy and non-linearities existed in
the relationships between output velocities and input voltages. Therefore, neural networks were considered to overcome the difficulties in the MR control.

6. Inverse neural network controller design

As mentioned above, due to the uncertainties and disturbances, a complete knowledge of the dynamics of a real MR is not attainable. Therefore, a NN controller was used to approximately model this function. It requires no prior information about the dynamics of the MR, as the NN learns them on-line or off-line.

As shown in Fig. 13, a neural network inverse model of the dc motor was directly attached to the motor as an inverse controller, where \( w_m \) and \( w \) denote the input angular velocity of the NN controller and the output angular velocity of the DC motor respectively, \( q_m \) and \( q \) denote the input displacement of the NN controller and the output displacement of the DC motor respectively, \( G \) denotes the gear ratio, and \( k_g \) is the gain factor. Five input variables were used: the velocity error, displacement error, current velocity and displacement, current input voltage of the motor. All these inputs were measured in one sampling period. The output of the NN was a predicted
The change of the input voltage for the next sampling period. The output voltages were summed up; therefore, a limiter was used to avoid integration wind-up.

The NN was trained off-line. All the training data were collected in open loop state. The collected data were randomly mixed to form an expanded data set. Ninety percent of the data were used for training and 10% for validation. A backpropagation learning with momentum and adaptive learning rate were used as the training algorithm. After training, two 5-10-1 neutral networks were obtained for the two dc motors, respectively.

7. Experimental results

Firstly, the performance of the proposed NN controller was compared to that of a PID controller by following a reference velocity with acceleration, steady and deceleration phases without using the tracking controller. Fig. 14 gives the comparison results of one wheel during these phases by using the NN controller and the PID controller, which was optimised by manual tuning. Fig. 14(a) is the comparison of displacement errors. Fig. 14(b) is the comparison of angular velocity errors.
Fig. 10. (a) Reference and actual trajectories (simulation); (b) reference and control velocity for square tracking (simulation); (c) square tracking errors under different disturbances (simulation).

It shows that the NN controller followed the reference velocity with less error than the PID controller, especially during the transient velocity phase of the servo.

Two types of experiments have been conducted with the experimental low-quality MR system. In one set of experiments the reference trajectory is a straight line with an initial error of 2 cm.
Fig. 10. Continued.
In the other set of experiments, a 0.6 by 0.6 m square path, which is comprised of circle or straight lines, was used as the reference trajectory (see Fig. 15). There was no initial error in the second set of experiments.

In the first set of experiments, although varying reference velocities ranging from 0 to 0.3 m/s were used, a final motion accuracy of better than 0.9 cm was achieved.

Fig. 16 shows the pattern profile of the reference velocity for the square trajectory. The MR
was decelerated from the maximum speed of 0.12 m/s to 0.06 m/s before turning a corner, and it was again accelerated to 0.12 m/s after turning the corner. The tracking controller parameters were set to: $k_1 = 10$, $k_2 = 2$, $\alpha = 4$. The total time taken for the trajectory was 38 s. The vehicle did not stop exactly at the start point, but a certain error remained. Fig. 17 gives the absolute position error as a function of time. It shows that during the entire trajectory the tracking position error remained below 1 cm.
8. Conclusions

A motion controller based on an inverse neural network controller and tracking controller using backstepping technique is proposed in this paper. The tracking controller is simulated under various situations, such as initial tracking error, linear velocities, different perturbations, white noise, Coulumbic friction and deadband, as may occur in a real MR. These simulations confirm the robustness of the proposed tracking controller. Another advantage is that the computation of the tracking control law (Eq. (13)) is very simple, and therefore suitable to be used in the real-time control applications.
The simulation model was based on the kinematics of an ideal MR, although disturbances were considered. Perfect knowledge of a real MR is unattainable, therefore, the NN controller, which does not require an exact mathematics model of the MR, shows the advantage of controlling the non-ideal MR. As the NN controller is used for the wheel-level control of the MR, it can generate a different tracking behaviour by redefining the feedback control law.

Experimental results on a low-quality mobile robot, which was built from cheap drivelines, show that (1) the neural networks can be used as an effective inverse controller of a low-quality de servo of a MR, which are highly tolerant to the noisy signals from the low-quality servo; (2) the integrated tracking controller improves the tracking performance of the experimental low-quality MR.

References