TRAJECTORY TRACKING CONTROL BASED ON ADAPTIVE NEURAL DYNAMICS FOR FOUR-WHEEL DRIVE OMNI-DIRECTIONAL MOBILE ROBOTS

J. Wang[1]– J. Chen[2*]– S. Ouyang[3]– Y. Yang[1]

[1]School of Automation, Guangdong University of Technology, Guangzhou 510090, Guangdong, China
[2]School of Computer Science, Zhaoqing University, Zhaoqing 526061, Guangdong, China
[3]School of Electronic Information and Mechnical & Electrical Engineering, Zhaoqing University, Zhaoqing 526061, Guangdong, China

ARTICLE INFO

***Article history*:**
Received: 24.01.2014.
Received in revised form: 05.03.2014.
Accepted: 10.03.2014.

Keywords:
Four-wheel drive
Omni-directional mobile robot
Adaptive neural dynamics
Trajectory tracking

Abstract:

There is usually the speed jump problem existing in conventional back-stepping tracking control for four-wheel drive omni-directional mobile robots, a trajectory tracking controller based on adaptive neural dynamics model is proposed. Because of the smoothness and boundedness of the output from the neural dynamics model, it produces a gradually varying tracking speed instead of the jumping speed, and the parameters are designed to avoid the control values exceeding their limits. And then, a parameter adaptive controller is presented to improve control performance. Simulation results of different paths and comparison with the conventional back-stepping technique show that the approach is effective, and the system has a good performance with smooth output.

1 Introduction

Omni-directional mobile robot can move in any direction without changing any position and pose because it has the character of omni-directional mobility [1]. With its special motion advantage, the omni-directional mobile robot is widely applied to the human production and life practice in recent years [2].The control problems of motion and regulation have been extensively studied in the field of omni-directional mobile robotics [3]. As an important part of that, the trajectory tracking control problem has also attracted the interest of many control researchers [4].

It is a common practice in mobile robotics to address control problems taking into account a kinematical representation [5]. From a kinematical perspective, the regulation and trajectory tracking control problems for the omni-directional mobile robot, have received sustained attention. Considering, only its kinematical model, many control strategies, like optimum control, robust control, sliding mode control, intelligent control, back-stepping control and so on, have been proposed [6-10]. For different virtual feedback values, different controllers based on back-stepping control could be designed, and they were also stably by using the back-stepping control approach. It is designed a back-stepping controller with global stability based on Newton mechanics model [10]. In literature [11] they solved the trajectory tracking problem with a nonlinear back-stepping controller for a three-wheel omni-directional mobile robot, and the control values were optimized by the sum of

* Corresponding author. Tel.: 0758-2752208; fax:0758-2181912;
Email: jpchen@zqu.edu.cn

squares technology. For the trajectory tracking control of an omni-directional wheeled robot for lower limbs rehabilitative training, in [12] the control problem and interference rejection are translated into L2 control design problem, and a tracking controller is presented by considering the back-stepping strategy. The achievements in these literatures are all usually gotten based on the back-stepping approach, however, when the initial tracking errors were bigger or the trajectory was discrete, there was always the speed jump existing in conventional back-stepping tracking control. It means that the acceleration or driving torque of the robot is big enough, even unlimited, which is impossible in practical application. Some researchers had solved the problem by neural dynamics model [13-16]. A neural dynamics mode is integrated with back-stepping approach to handle the jump-problem between speed and torque, which are the control outputs in the path tracking controller for a wheeled mobile robot [14]. Because of the properties of the model, the control signal is smooth and bound. But the control signals will exceed their limits if the parameters of neural dynamics model are not suitable. In literature [15], to deal with the speed-jump in the conventional back-stepping tracking control for underwater vehicles, they propose a hybrid control combining the back-stepping and the sliding-mode control based on the biological inspired model. Because that the control signal is limited, the control performances, like response speed and tracking accuracy, are usually getting worse.

We address and solve the trajectory tracking control problem of a four-wheel drive omni-directional mobile robot (FDOMR) taking into account its kinematical model. A tracking control approach based on adaptive neural dynamics model (ANDM) is presented. The stability of the closed-loop system and the convergence of the adapting process, are all strictly guaranteed. For different trajectories, the effectiveness of the control scheme is demonstrated through simulation study, and is compared with the back-stepping control and traditional neural dynamics model. It is shown that the control system with the ANDM has better tracking performance, without any jumping speed.

2 Model of FDOMR

2.1 Kinematical model of FDOMR

Four-wheel robots are one of the models of robots, which are used in many domains. They are omni-directional with four wheels that have the ability of moving to any direction at any time (they are holonomic mobile robots, in other words). Figure 1 shows the schematic of a four-wheel robot, the angles and directions of the four wheels.

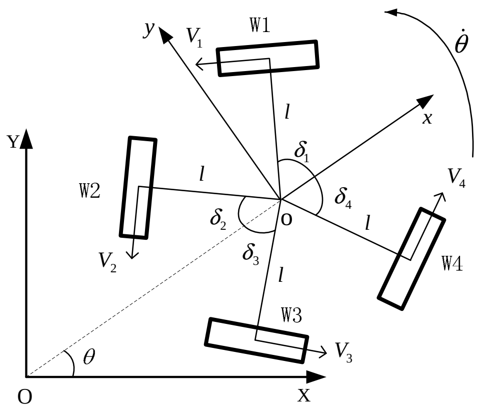


Figure 1. Wheel placement of the FDOMR.

Where *XOY* is the world coordinate frame for robot, *xoy* is the robot own coordinate frame, θ denotes the moving direction of robot, $W_i(i=1, 2, 3, 4)$ denotes every wheel, δ_i denotes the angle between wheel and x axis respectively, V_i denotes the velocity of each wheel, its positive direction is anticlockwise, l is the distance between the center of robot-body and that of each wheel.

According to the geometric relationship of Fig. 1, suppose that the postures (position and orientation) of robot in its own coordinate frame and the world coordinate frame are separately expressed as $[x\ y\ \varphi]^T$ and $[X\ Y\ \psi]^T$, the velocity vector in the own coordinate frame is $[u\ v\ \omega]^T = [\dot{x}\ \dot{y}\ \dot{\varphi}]^T$ .The kinematical equation of mobile robot can be constructed as:

$$\begin{cases} \dot{X} = u\cos\theta - v\sin\theta, \\ \dot{Y} = u\sin\theta + v\cos\theta, \\ \dot{\psi} = \omega. \end{cases} \tag{1}$$

In Eq. (1) $\theta = \psi - \varphi$.

2.2 Trajectory tracking error model of FDOMR

Considering the reference postures in the world coordinate frame and robot own coordinate frame are $[X_r\ Y_r\ \psi_r]^T$ and $[x_r\ y_r\ \varphi_r]^T$, and then the reference velocities are $[\dot{X}_r\ \dot{Y}_r\ \dot{\psi}_r]^T$ and $[u_r\ v_r\ \omega_r]^T$.
In the world coordinate frame and the robot own coordinate frame, defining the errors between the reference posture and the real posture as $\varepsilon^T = [X_e\ Y_e\ \psi_e]^T$ and $e^T = [e_x\ e_y\ e_\varphi]^T$, then we have the relationship between them as:

$$\begin{bmatrix} e_x \\ e_y \\ e_\varphi \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_e \\ Y_e \\ \psi_e \end{bmatrix} \tag{2}$$

where $\varepsilon^T = [X_r - X\quad Y_r - Y\quad \psi_r - \psi]^T$, and $e^T = [x_r - x\quad y_r - y\quad \varphi_r - \varphi]^T$.
Introducing Eq. (2) into Eq. (1), we have the model of kinematical error:

$$\begin{cases} \dot{e}_x = -u + v_p \cos\varphi + \omega e_y \\ \dot{e}_y = -v + v_p \sin\varphi - \omega e_x \\ \dot{e}_\varphi = -\omega + \omega_r \end{cases} \tag{3}$$

where $v_p = \sqrt{\dot{X}_r^{\,2} + \dot{Y}_r^{\,2}} = \sqrt{u_r^{\,2} + v_r^{\,2}}$.
From Eq. (2) we obtain that the trajectory tracking error in robot own coordinate frame $\lim_{t\to\infty} \left\| e^T(t) \right\| = 0$, meanwhile, the error in the world coordinate frame $\lim_{t\to\infty} \left\| \varepsilon^T(t) \right\| = 0$. Considering the robot control system described as Eq. (1), the tracking error in closed loop can be globally asymptotically stabilized to zero with the suitable undetermined control law $U = [u\ \ v\ \ \omega]^T$.

3 Trajectory tracking controller based on adaptive neural dynamics model

3.1 Back-stepping based trajectory tracking controller

Simple back-stepping approach has been widely applied to the motion control of mobile robots, based which the controller can be designed with stable performance. Taking now the following Lyapunov function candidate as $V_1 = \frac{1}{2}(e_x^{\,2} + e_y^{\,2} + e_\varphi^{\,2})$, considering Eq. (3), the time derivative is given as:

$$\begin{aligned} \dot{V}_1 &= e_x \dot{e}_x + e_y \dot{e}_y + e_\theta \dot{e}_\theta \\ &= e_x(-u + v_p \cos\varphi + \omega e_y) \\ &\quad + e_y(-v + v_p \sin\varphi - \omega e_x) + e_\varphi(\omega_r - \omega) \\ &= e_x(-u + v_p \cos\varphi) + e_y(-v + v_p \sin\varphi) \\ &\quad + e_\varphi(\omega_r - \omega) \end{aligned} \tag{4}$$

According to the theory of Lyapunov stability, we can obtain the back-stepping control law,

$$U = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1 e_x + \dot{x}_r \cos\theta + \dot{y}_r \sin\theta \\ k_2 e_y - \dot{x}_r \sin\theta + \dot{y}_r \cos\theta \\ k_3 e_\varphi + \omega_r \end{bmatrix} \tag{5}$$

where k_1, k_2, k_3 are the positive constants.

3.2 Model of neural dynamics

Neural dynamic model is the cytomembrane film model in the study process of biology nervous system, which is gotten by using circuitry theory to research the biology film. It describes the real-time adaptive behavior of individuals to complex and dynamic environment contingencies and has been applied in many areas, such as biological, machine vision, and robotics and so on. Based on the traditional neural dynamics model, a typical simple model is given by Yang [13], which is described as:

$$d\xi_i / dt = -A\xi_i + (B - \xi_i)S_i^+ - (D + \xi_i)S_i^- \tag{6}$$

where $\xi_i(t)$ is the membrane potential of the i neuron. A denotes the passive decay rate. B and D are the upper and lower bounds of the membrane potential. S_i^+ and S_i^+ are excitatory and inhibitory inputs, respectively, which are defined as:

$$S_i^+(x(t)) = \max(0, x),\ S_i^-(x(t)) = \max(-x, 0) \tag{7}$$

When $x>0$, considering $\xi_i(t_0)=0$ and Eq. (6), we have:

$$\xi_i(t) = \frac{Bx}{A + x}(1 - e^{-(A+x)t}) \tag{8}$$

If $x(t)\to\infty$, further we have:

$$\lim_{x\to\infty}\xi_i(t)=\lim_{x\to\infty}\frac{Bx}{A+x}=B \tag{9}$$

In the same way, when $x<0$, $\lim_{x\to\infty}\xi_i(t)=-D$.

Even though the signal $x(t)\to\infty$ the output $\xi_i(t)$ of neural dynamics model described by Eq. (6) is still limited in the area [-D,B].

3.3 Trajectory tracking controller based on neural dynamics model

Considering the actual application, the Eq. (6) can be rewritten as:

$$dE_i/dt=-A_iE_i+(B_i-E_i)f(e_i)-(D_i+E_i)g(e_i) \tag{10}$$

where $f(e_i)=\max(0,e_i)$, $g(e_i)=\max(-e_i,0)$, $e_i, i=x,y,\varphi$ are the tracking errors in robot own coordinate frame.

At the initial moment, supposing that the tracking error $e_i(0)\neq 0$, and when $e_i(0)>0$, thinking about Eq. (8) and Eq. (10), we have:

$$E_i(t)=\frac{B_ie_i(0)}{A_i+e_i(0)}(1-e^{-(A_i+e_i(0))t}) \tag{11}$$

Then, $E_i(0)=0$. In other words, the Eq. (10) can actively restrain the jumping speed caused by the initial errors.

Using E_i instead of the tracking error e_i and taking it into Eq. (5), we have the trajectory tracking control law based on neural dynamics model described as:

$$U=\begin{bmatrix}u\\v\\\omega\end{bmatrix}=\begin{bmatrix}k_1E_x+\dot{X}_r\cos\theta+\dot{Y}_r\sin\theta\\k_2E_y-\dot{X}_r\sin\theta+\dot{Y}_r\cos\theta\\k_3E_\varphi+\omega_r\end{bmatrix} \tag{12}$$

From Eq. (12), if the parameters were not suitable for the control system, even though the outputs of the neural dynamics model have been limited in the bounds [-D_i,B_i]., but the out-of-limit potential for existential risk is not eliminated. Then the parameters should follow the rules described as Eq. (13) and Eq. (14).

$$\begin{cases}B_x\le(u_{\max}-|\dot{X}_{r\max}|-|\dot{Y}_{r\max}|)/k_1\\B_y\le(v_{\max}+|\dot{X}_{r\max}|-|\dot{Y}_{r\max}|)/k_2\\B_\varphi\le(\omega_{\max}-|\omega_{r\max}|)/k_3\end{cases} \tag{13}$$

$$\begin{cases}D_x\le-(u_{\min}-|\dot{X}_{r\min}|-|\dot{Y}_{r\min}|)/k_1\\D_y\le-(v_{\min}+|\dot{X}_{r\min}|-|\dot{Y}_{r\min}|)/k_2\\D_\varphi\le-(\omega_{\min}-|\omega_{r\min}|)k_3\end{cases} \tag{14}$$

where $[u_{\max}\ v_{\max}\ \omega_{\max}]^T$ and $[u_{\min}\ v_{\min}\ \omega_{\min}]^T$ are the upper and lower bounds of outputs, $[\dot{X}_{r\max}\ \dot{Y}_{r\max}\ \omega_{r\max}]^T$ and $[\dot{X}_{r\min}\ \dot{Y}_{r\min}\ \omega_{r\min}]^T$ are the upper and lower bounds of desired velocities.

3.4 Adaptive parameters for neural dynamics model

Because of the smoothness and boundedness of the output from the neural dynamics model, the output is limited in its area, and some performances, like response speed and tracking accuracy, is becoming worse. To solve this effect and improve the control performance, we redesign the model parameters with self-adaptation.

Taking $\hat{E}_i$ is the estimated value of E_i, and then we have:

$$d\hat{E}_i/dt=-\hat{A}_i\hat{E}_i+(\hat{B}_i-\hat{E}_i)f(e_i)-(\hat{D}_i+\hat{E}_i)g(e_i) \tag{15}$$

where $i=x,y,\varphi$, $\hat{A}_i$, $\hat{B}_i$, $\hat{D}_i$ are the estimated values of parameters A_i, B_i, D_i, and their adaptive laws are:

$$\dot{\hat{A}}_i=\alpha_i\hat{E}_i(\hat{E}_i-E_i),\ \ \alpha_i>0, \tag{16}$$

$$\dot{\hat{B}}_i=\begin{cases}\alpha_ie_i(E_i-\hat{E}_i) & e_i>0\\0 & e_i\le 0\end{cases}, \tag{17}$$

$$\dot{\hat{D}}_i=\begin{cases}\alpha_ie_i(E_i-\hat{E}_i) & e_i<0\\0 & e_i\ge 0\end{cases}. \tag{18}$$

Defining the estimated errors of E_i is $\tilde{E}_i=E_i-\hat{E}_i$, thinking about Lyapunov function as:

$$V_2 = \frac{1}{2}\tilde{E}_i^2 + \frac{1}{2\alpha_i}(A_i - \hat{A}_i)^2 + \frac{1}{2\alpha_i}(B_i - \hat{B}_i)^2 + \frac{1}{2\alpha_i}(D_i - \hat{D}_i)^2 \quad (19)$$

And then, we obtain:

$$\dot{V}_2 = \tilde{E}_i\dot{\hat{E}}_i - \frac{1}{\alpha_i}[(A_i - \hat{A}_i)\dot{\hat{A}}_i + (B_i - \hat{B}_i)\dot{\hat{B}}_i + (D_i - \hat{D}_i)\dot{\hat{D}}_i] \quad (20)$$

When $e_i > 0$, $f(e_i) = e_i$, $g(e_i) = 0$, $\dot{\hat{D}}_i = 0$, from Eq. (10), Eq. (16), Eq. (17) and Eq. (18):

$$\begin{aligned}\dot{V}_2 &= (E_i - \hat{E}_i)(\dot{E}_i - \dot{\hat{E}}_i) - \frac{1}{\alpha_i}[(A_i - \hat{A}_i)\alpha_i\hat{E}_i(\hat{E}_i - E_i)] \\ &\quad - \frac{1}{\alpha_i}[(B_i - \hat{B}_i)\alpha_i e_i(E_i - \hat{E}_i)] \\ &= (E_i - \hat{E}_i)[-A_iE_i + (B_i - E_i)f(e_i) + \hat{A}_i\hat{E}_i - (\hat{B}_i - \hat{E}_i)f(e_i)] \\ &\quad - [(A_i - \hat{A}_i)\hat{E}_i(\hat{E}_i - E_i) + (B_i - \hat{B}_i)e_i(E_i - \hat{E}_i)] \\ &= (E_i - \hat{E}_i)[-A_iE_i + \hat{A}_i\hat{E}_i + (B_i - E_i - \hat{B}_i + \hat{E}_i)e_i] \\ &\quad + (A_i - \hat{A}_i)\hat{E}_i(E_i - \hat{E}_i) - (B_i - \hat{B}_i)e_i(E_i - \hat{E}_i) \\ &= (E_i - \hat{E}_i)[-A_i(E_i - \hat{E}_i) - (E_i - \hat{E}_i)e_i] \\ &= -(A_i + e_i)(E_i - \hat{E}_i)^2\end{aligned} \quad (21)$$

As $A_i > 0$ and $e_i > 0$, $-(A_i + e_i) < 0$, then we have $\dot{V}_2 < 0$. In the similar way $\dot{V}_2 < 0$, when $e_i < 0$. It means that the adaptive laws Eq. (16)-(18) can make the estimated errors convergent to zero in limited time.

The final trajectory tracking control law based on neural dynamics model is described as:

$$U = \begin{bmatrix} u \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} k_1\hat{E}_x + \dot{X}_r\cos\theta + \dot{Y}_r\sin\theta \\ k_2\hat{E}_y - \dot{X}_r\sin\theta + \dot{Y}_r\cos\theta \\ k_3\hat{E}_\varphi + \omega_r \end{bmatrix} \quad (22)$$

4 Stability analysis of control system

Considering the Eq. (2) in closed loop with the new control law Eq. (22) and the adaptive law Eq. (16)-(18), we have:

$$V = \frac{1}{2}(\hat{B}e_x^2 + \hat{B}e_y^2 + k_1\hat{E}_x^2 + k_2\hat{E}_y^2) + \frac{1}{2}(\hat{B}e_\varphi^2 + k_3\hat{E}_\varphi^2) \quad (23)$$

Assuming that $V_3 = \frac{1}{2}(\hat{B}e_x^2 + \hat{B}e_y^2 + k_1\hat{E}_x^2 + k_2\hat{E}_y^2)$, and $V_4 = \frac{1}{2}(\hat{B}e_\varphi^2 + k_3\hat{E}_\varphi^2)$, Eq. (23) can be rewritten as:

$$V = V_3 + V_4 \quad (24)$$

Then we have their time derivatives as:

$$\begin{aligned}\dot{V}_3 &= \hat{B}e_x\dot{e}_x + \hat{B}e_y\dot{e}_y + k_1\hat{E}_x\dot{\hat{E}}_x + k_2\hat{E}_y\dot{\hat{E}}_y \\ &= -\hat{B}k_1e_x\hat{E}_x - \hat{B}k_2e_y\hat{E}_y + k_1[-\hat{A} - f(e_x) \\ &\quad - g(e_x)]\hat{E}_x^2 + k_2[-\hat{A} - f(e_y) - g(e_y)]\hat{E}_y^2 \\ &\quad + k_1[\hat{B}f(e_x) - \hat{D}g(e_x)]\hat{E}_x + k_2[\hat{B}f(e_y) \\ &\quad - \hat{D}g(e_y)]\hat{E}_y \\ &= k_1[-\hat{A} - f(e_x) - g(e_x)]\hat{E}_x^2 + k_1[\hat{B}f(e_x) \\ &\quad - \hat{D}g(e_x) - \hat{B}e_x]\hat{E}_x + k_2[-\hat{A} - f(e_y) \\ &\quad - g(e_y)]\hat{E}_y^2 + k_2[\hat{B}f(e_y) - \hat{D}g(e_y) - \hat{B}e_y]\hat{E}_y\end{aligned} \quad (25)$$

$$\begin{aligned}\dot{V}_4 &= \hat{B}e_\varphi\dot{e}_\varphi + k_3\hat{E}_\varphi\dot{\hat{E}}_\varphi \\ &= -\hat{B}k_3e_\varphi\hat{E}_\varphi + k_3[-\hat{A} - f(e_\varphi) - g(e_\varphi)]\hat{E}_\varphi^2 \\ &\quad + k_3[\hat{B}f(e_\varphi) - \hat{D}g(e_\varphi)]\hat{E}_\varphi \\ &= k_3[-\hat{A} - f(e_\varphi) - g(e_\varphi)]\hat{E}_\varphi^2 \\ &\quad + k_3[\hat{B}f(e_\varphi) - \hat{D}g(e_\varphi) - \hat{B}e_\varphi]\hat{E}_\varphi\end{aligned} \quad (26)$$

With the definitions of $f(e_i)$ and $g(e_i)$, we have $\hat{B} = \hat{D}$. When $e_i > 0$, $f(e_i) = e_i$, $g(e_i) = 0$, taking them into Eq. (25) and Eq. (26), we get $\hat{A} + f(e_i) + g(e_i) > 0$, and $\hat{B}f(e_i) - \hat{D}g(e_i) - \hat{B}e_i = 0$, we have $\dot{V}_3 < 0, \dot{V}_4 < 0$. Similarly, when $e_i < 0$, we also have $\dot{V}_3 < 0, \dot{V}_4 < 0$.

In conclusion, $\dot{V} < 0$, it means that the control system is globally asymptotically stabilized.

5 Simulation and analysis

We carried out numerical simulations at Matlab 7.0 platform to assess the performance of the controller given in Eq. (22). The control parameters correspond to a laboratory prototype built in our institution and they are found as follows.

The speed constraints of our mobile robot are $|u| \le 3m/s$, $|v| \le 3m/s$ and $|\omega| \le 3rad/s$, sampling time is 0.01 s, parameters of back-stepping controller are $k_1 = 1$, $k_2 = 1$, $k_3 = 3$, initial values of the adaptive neural dynamics model are $\hat{A}_i(0) = 4$, $\hat{B}_i(0) = 4$, $\hat{D}_i(0) = 4$, and its adaptive parameters are $\alpha_x = 0.3$, $\alpha_y = 0.5$, $\alpha_\varphi = 0.1$.

5.1 Tracking results and analysis for different trajectories

To assess the performance of our control law, we conducted some experiments of tracking the straight path and circle path with initial errors respectively.

a. straight path

For tracking the straight path, its equation is $Y_r = X_r$, angle is $\varphi_r = \pi/4$, the desired line velocities of robot are $u_r = 0.5m/s$, $v_r = 0.5m/s$, the desired angle velocity is $\omega_r = 0rad/s$, actual initial posture is $(1,-1,\pi/2)^T$, and its reference posture is $(0,0,0)^T$, then the initial tracking error is $(-1,1,-\pi/2)^T$. The simulation results are shown on Fig. 2.

Figure 2(a) and (d) shows that the robot can track the straight path and correct deviations quickly (about 3 s). From Fig. 2 (b) and (c), at $t = 0$, even if the initial error is $(-1,1,-\pi/2)^T$, the output of the controller is still zero.

And the maximum of real line velocities and angle velocity are $u = 0.68m/s$, $v = 0.69m/s$, $\omega = 1.6rad/s$, which are much smaller than their upper bounds.

b. circle path

When the robot was tracking the circle path, its equation is given as $X_r = r\cos t$, $Y_r = r\sin t$, $\varphi_r = t$, where the radius $r = 1m$, $0 \le t \le 15$. At the initial moment $t = 0s$, the actual initial posture is $(2, 2.5, \pi/2)^T$, and the reference posture is $(1,0,0)^T$, therefore the initial tracking error is $(-1,-2.5,-\pi/2)^T$. The simulation results are shown on Fig. 3.

From Fig. 3(a), it shows that the robot can track the time varying path with a good performance. From Fig. 3(d), it can be observed that the controller can correct posture errors quickly. In Fig. 3 (b) and (c), the actual line velocities and angle velocity are also within their upper bounds.

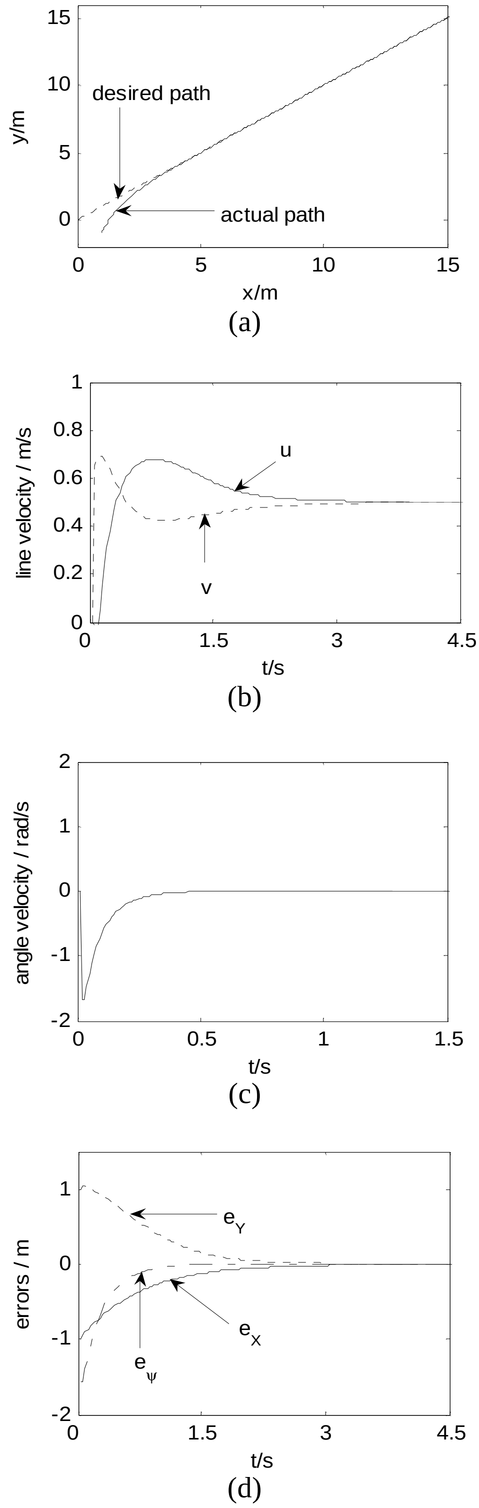
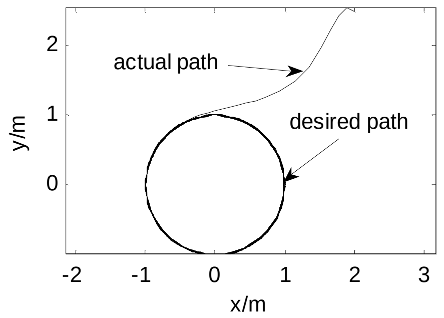
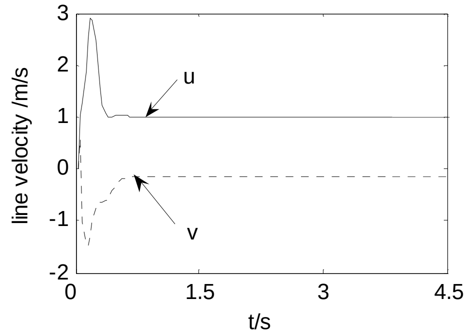


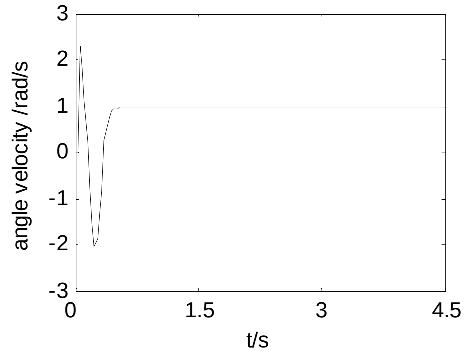
Figure 2. Tracking results for straight path: (a) tracking result, (b) line velocity, (c) angle velocity, (d) posture errors.



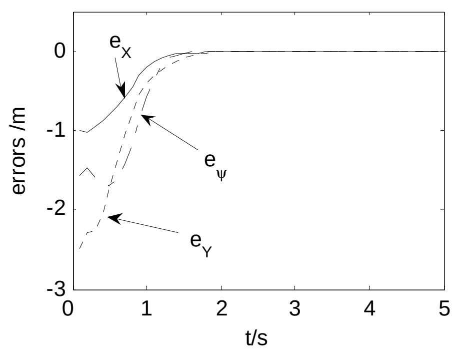
(a)



(b)



(c)



(d)

Figure 3. Tracking results for circle path: (a) tracking result, (b) line velocity, (c) angle velocity, (d) posture errors.

5.2 Comparison with other approaches

Further more, to test the performance of our control law, we let the robot tracking a straight path with some controllers, they are the controller based on back-stepping approach (BAC) [11], the controller based on traditional neural dynamics model (TNDMC) with different parameters, and our adaptive neural dynamics model based controller (ANDMC). The parameters of BAC and ANDMC are the same as mentioned above, one of the TNDMC named TNDMC1, whose parameters are the bigger values, $A=10$, $B=10$, $D=15$, and the other is TNDMC2, its parameters are the smaller values, $A=0.5$, $B=0.5$, $D=0.5$, which are as same as that in literature [15]. The simulation results are shown on Fig. 4 and Fig. 5.

Comparing with Fig. 2(d), Fig. 4(a) and (d) show that the robot can track the path more quickly with BAC than ANDMC. It is because that the BAC can product a large output, which has already exceeded the speed bounds. From Fig. 4(c) and (d), because of the large errors at initial moment, the velocities alter from zero to large value quickly, and then they reduce towards opposite direction. This action is done repeatedly until the velocities reach the proper values. From Fig. 5, the response speed of trajectory tracking with TNDMC1 is the most quickly, but that with TNDMC2 is the slowest. However, the smoothness of velocity tracking is oppositely. The velocities of TNDMC1 have a wide shake and have exceeded their bounds far away, but the smoothness of velocities with TNDMC2 is the best. This is because that the bigger parameters can make the response speed of trajectory tracking faster, however, they also make the outputs exceed their bounds easier. As using the adaptive approach, ANDMC can improve the response speed and tracking accuracy.

6 Conclusion

The trajectory tracking control with kinematical model for a four-wheel omni-directional mobile robot has been addressed and solved by means of an adaptive neural dynamics model. The character of neural dynamics model has been applied to smooth the output for conquering the speed jump problem, and a new strategy for confirming the values of parameters has been given. An adaptive control law has been presented to improve the response speed

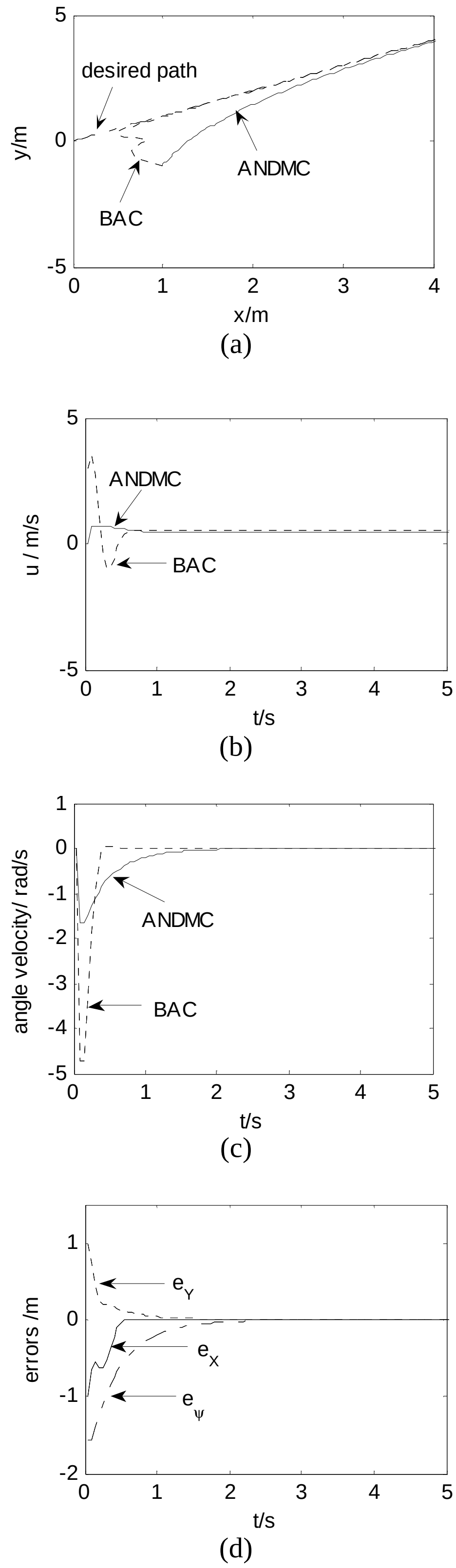


Figure 4. Simulation results compared with back-stepping approach: (a) tracking result, (b) line velocity u, (c) angle velocity, (d) posture errors with BAC.

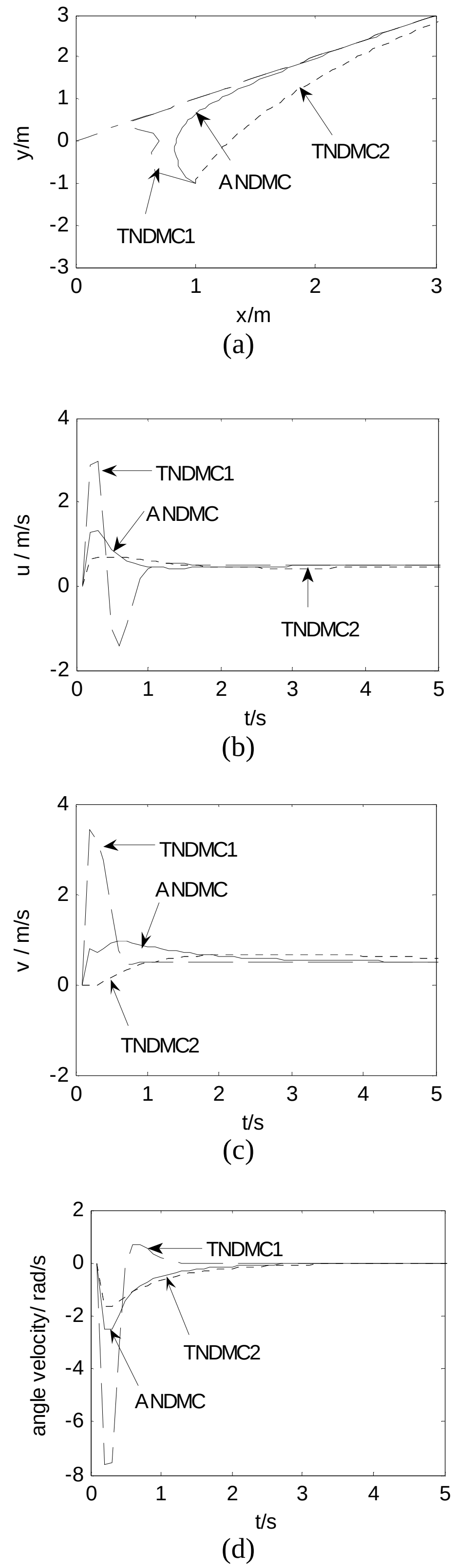


Figure 5. Simulation results compared with other neural dynamics models: (a) tracking result, (b) line velocity u, (c) line velocity v, (d) angle velocity.

and tracking accuracy. Using the Lyapunov theory,the controller scheme has been demonstrated to be stable. All the simulation results have indicated that the proposed control strategy is effective to solve the path tracking problem.

Acknowledgements

This work of this paper was supported by the Natural Science Fund Free Application Project of Guangdong Province (No. S2011010004006), the Industry-university-institute Cooperation Project of Guangdong Province and China's Ministry of Education (No. 2012B091100423), the Science and Technology Planning Project of Zhaoqing City (No. 2010F006, No. 2011F001), and the Research Initiation Fund of Zhaoqing University (No. 2012BS01).

References

[1] Jefri E. M., Mohamed R., Sazali Y.: *Designing omni-directional mobile robot with mecanum wheel*, American Journal of Applied Sciences, 3 (2006), 5, 1831-1835.

[2] Liu Y., et al.: *Omni-directional mobile robot controller design by trajectory linearization*, Proceedings of the 2003 American Control Conference, USA, 2003, 3423-3428.

[3] Williams R. L., Carter B. E., Gallina P., et al.: *Dynamic model with slip for wheeled omnidirectional robots*, IEEE Transactions on Robotics and Automation, 18 (2002), 3, 285-293.

[4] Kalmar-nagy T., Raffaello D. A., Gangguly P.: *Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle*, Robotics and Autonomous System, 46 (2004), 5, 47-64.

[5] Purwin O., Andrea R. D.: *Trajectory generation and control for four wheeled omnidirectional vehicles*, Robotics and Autonomous System, 54 (2006), 13-22.

[6] Xiong R., Zhang H., Chu J., et al.: *Modeling and optimal control of omni-derectional mobile robots*, Control Theory and Applications, 23 (2006), 1, 96-98.

[7] Huang H. C.: *FPGA implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform*, Industrial Electronics, IEEE Transactions on, 56 (2009), 5, 1604-1616.

[8] Zhao D. B., Yi J. Q., et al.: *Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network-based sliding mode approach*, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39 (2009), 3, 788-799.

[9] Wong C. C., Lin Y. H., Lee S., et al.: *Ga-based fuzzy system design in FPGA for an omni-directional mobile robot*, Journal of Intelligent and Robotic Systems, 44 (2005), 5, 327-347.

[10] Krajči, V., Stojković, N.: *Improvement of robot trajectory tracking by using nonlinear control methods*, Engineering Review, 26 (2006), 1, 9-17.

[11] Chen S., Juang J. C., Su S. H.: *Backstepping control with sum of squares design for omni-directional mobile robots*, Proceedings of ICROS-SICE International Joint Conference, Hakata-ku, Fukuoka City, Japan, 2009, 545-550.

[12] Yang J. Y., Bai D. C., Wang S. Y., et al.: *Trajectory tracking control of omni-directional wheeled robot for lower limbs rehabilitative training*, ROBOT, 33 (2011), 3, 314-318.

[13] Yang H., Yang S. X., Mittal G. S.: *Tracking control of a nonholonomic mobile robot by integrating feedback and neural dynamics techniques*, Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV ,USA, 2003, 3522-3527.

[14] Cao Z. C., Zhao Y. T., Wu Q. D.: *Point stabilization of a nonholonomic mobile robot based on backstepping and neural dynamics*, Acta Electronica Sinica, 39 (2011), 3, 591-595.

[15] Sun B., Zhu D. Q., Deng Z. G.: *Bio-inspired discrete trajectory-tracking control for open-frame underwater vehicles*, Control Theory and Applications, 30 (2013), 4, 454-462.

[16] Yang S. X., Zhu A., Yuan G. F., et al.: *A bioinspired neurodynamics based approach to tracking control of mobile robots*, IEEE Transactions on Industrial Electronics, 59 (2012), 8, 3211-3220.