The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Trajectory Tracking and Point Stabilization of Noholonomic Mobile Robot

Zhengcai Cao, Yingtao Zhao, and Shuguo Wang

Abstract—**In this paper, a mixed controller for solving the trajectory tracking and point stabilization problems of a mobile robot is presented, applying the integration of backstepping technique and neural dynamics. By introducing a virtual target point, the whole motion process is divided into two parts. The first one is employed to realize tracking control and the other one is adopted to implement point stabilization. Each part produces a feedback control law by using backstepping technique. Moreover, to solve the speed and torque jump problems and make the controller generate smooth and continuous signal when controllers switch, the neural dynamics model is integrated into the backstepping. The stability of the proposed control system is analyzed by using Lyapunov theory. Finally, simulation results are given to illustrate the effectiveness of the proposed control scheme.**

I. INTRODUCTION

In recent decades, wheeled mobile robots have gained increasing attention both in the robotics and control areas, due to their broad and promising applications in the various industrial and service fields. Many researchers have worked in this field for a long period. Such studies have been divided into two main portions: tracking of a reference trajectory and stabilization to a fixed posture.

Trajectory tracking of mobile robots aims at controlling robots to track a given time varying trajectory. This problem has been addressed by sliding mode control technique [1], fuzzy scheme [2], model predictive [3], state feedback control [4] and adaptive control [5].These controllers require that linear or angular speeds must not converge to zero, i.e., reference trajectories are persistently excited.

The point stabilization can be regarded as the generation of control inputs to drive the robot from any initial point to a target point. Several methods have been applied to solve the stabilization problem. A model predictive control scheme was adopted in [6]. S. Li [7] proposed an adaptive control law to deal with the stabilization with unknown kinematic parameters. These control methods demand that the linear and angular velocities are zero at the target point.

In motion control of mobile robot, it is more realistic that the robot tracks along a reference trajectory and then parks at target point. It is difficult to solve this problem in a unified approach because of the different velocity requirement of tracking and stabilization. Therefore, the two problems are only studied separately by researchers.

In this paper, considering the kinematic and dynamic model, the tracking and stabilization problems of mobile robots are addressed by using backstepping technique and neural dynamics. The simulation results show that the presented controller can produce smooth and continuous control signal to guide the robot to track a reference trajectory and park at the target point with a quite small error.

This paper is organized as follows. Section II introduces the mobile robot model and neural dynamic model. In Section III, the problem of tracking and stabilization is stated. The tracking and stabilization control scheme is described in Section IV. Simulation results are provided in Section V. Finally, Section VI concludes the paper.

II. BACKGROUND

In this section, a mobile robot model and a neural dynamics model are briefly introduced.

A. Mobile Robot Model

The mobile robot with two independent driven wheels is shown in Fig.1. *O–XY* is the world coordinate system and *C–X'Y'* is the coordinate system fixed to the robot. The mass center *C* of robot is located in the middle of the driving wheels. r is the radius of rear wheels and l is the distance of rear wheels. m is the mass of the body and I is the moment of inertia of the body about the vertical axis through *C*.

The kinematic model for the mobile robot under the nonholonomic constraint of pure rolling and non-slipping is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = S(q)\bar{v}(t) \qquad (1)$$

where $q=[x,y,\theta]^T$ is actual posture of robot, (x,y) are the coordinates of *C*, θ is the orientation angle of robot. $S(q)\in R^{3\times 2}$ and $\bar{v}\in R^2$ represent the full rank velocity transformation matrix and velocity vector. v and w are linear and angular velocities of mobile robot, respectively.

Manuscript received June 15, 2010. This work was supported by State Key Laboratory of Robotics and System and State Key Lab of Embedded System and Service Computing.

Zhengcai Cao is with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China (e-mail: giftczc@163.com).

Yingtao Zhao is with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China (e-mail: yingtaozhao@sohu.com).

Shuguo Wang is with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin, China.

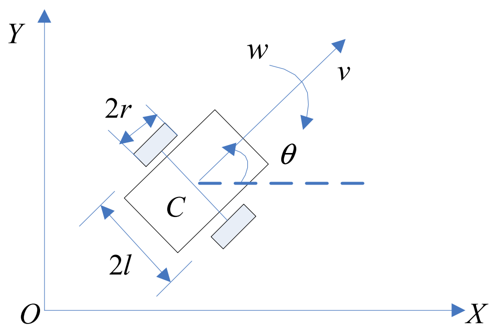


Fig. 1. Model of mobile robot.

Note that the mobile robot has the nonholonomic constraint, where the driving wheels roll purely and do not slip, i.e.

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

where $A(q) = [-\sin\theta \ \ \cos\theta \ \ 0]$.

The dynamic equation [5] of the simple model of the mobile robot, assuming all the uncertainties and disturbances are zero, can be described as

$$\overline{M}(q)\dot{\overline{v}} = \overline{\tau} \tag{3}$$

where $\overline{\tau} = [\tau_1, \tau_2]^T$ denote linear and angular torques, respectively. $\overline{M}$ and $\overline{B}$ are selected as

$$\overline{M} = \begin{bmatrix} m & 0 \\ 0 & I \end{bmatrix}, \ \ \overline{B} = \frac{1}{r}\begin{bmatrix} 1 & 0 \\ l & -l \end{bmatrix}.$$

B. Neural Dynamics Model

The neural dynamics model can depict the real-time adaptive behavior of individuals to complex and dynamic environment contingencies and has been applied in robotics. A typical dynamics model is described as [8]

$$d\xi_i / dt = -A\xi_i + (B - \xi_i)S_i^+ - (D + \xi_i)S_i^- \tag{4}$$

where ξ_i is the membrane potential of the ith neuron. A represents 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

$$S_i^+(x) = \max(0, x)$$
$$S_i^-(x) = \max(0, -x) \tag{5}$$

The neural dynamics characterized by (7) is restricted to a bounded interval [-*D*, *B*] for any excitatory and inhibitory inputs. The model is a continuous differential equation and the outputs are continuous and smooth. Its advantages such as stability and efficient computation are very attractive for motion control of robots.

III. PROBLEM STATEMENT

In general, the trajectory tracking problem aims at tracking a reference mobile robot with a known posture $q_r = [x_r, \ \ y_r, \ \ \theta_r]^T$ which is generated by

$$\begin{cases} \dot{x}_r = v_r \cos\theta_r \\ \dot{y}_r = v_r \sin\theta_r \\ \dot{\theta}_r = w_r \end{cases} \tag{6}$$

where v_r and w_r are the reference linear and angular velocities, respectively. θ_r is the reference angular.

Assumption 1. For the tracking problem, it is assumed that the reference velocities v_r and w_r do not go to zero simultaneously. That is, it is assumed that at any time either $\lim_{t\to\infty} v_r(t) \neq 0$ and/or $\lim_{t\to\infty} w_r(t) \neq 0$ [9].

We define the errors between the actual and reference posture as

$$\overline{q} = q_r - q = \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} \tag{7}$$

The posture error e_p expressed in the frame of the real robot, as shown in Fig.2, reads:

$$e_p = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = T_e(q_r - q) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} \tag{8}$$

where T_e is transformation matrix.

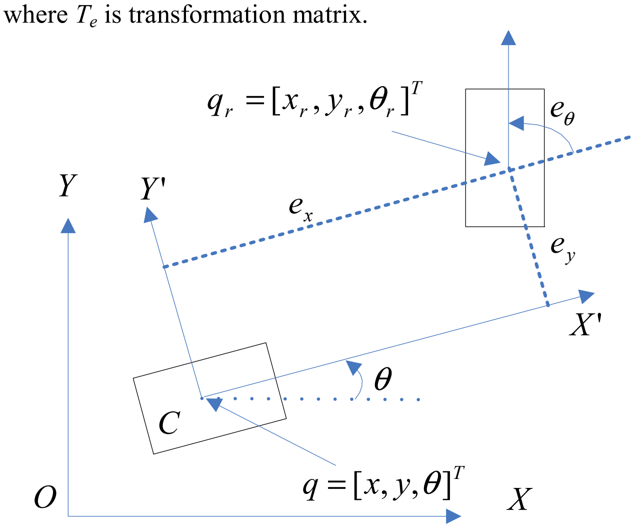


Fig. 2. Robot posture error coordinate.

Therefore, the tracking problem, under the Assumption 1, is to find a feedback control law to force the robot to track the reference trajectory precisely such that tracking error $e_p = [e_x, \ \ e_y, \ \ e_\theta]^T$ tends to zero.

In order to make the robot park at the final pose, a virtual target point $q_r(P)$ is introduced, which is located on the given trajectory and near the real target point $q_r(N)$. Herein, P and N denote the Pth and Nth time instant, respectively. When the robot reaches the virtual target, linear and angular velocities should reduce asymptotically and then converge to zero at the goal. The posture error can still be expressed by (8).

Assumption 2. For the point stabilization problem, it is assumed that the reference velocities v_r and w_r go to zero simultaneously. That is, it is assumed that at any time $\lim_{t\to\infty} v_r(t) = 0$ and $\lim_{t\to\infty} w_r(t) = 0$.

So point stabilization problem, under the Assumption 2, is to design a controller $\bar{\tau} = [\tau_1, \tau_2]^T$ for $\lim_{t\to\infty} e_q = 0$ and $\lim_{t\to\infty} \bar{\tau} = 0$.

From above analysis, the trajectory tracking and point stabilization problems to reference mobile robot can be stated as: find a control law so that the state (e_x, e_y, e_θ) can be held near the origin (0, 0, 0), and finally, the controller outputs can converge to zero.

IV. CONTROLLER DESIGN

In this section, considering the model of mobile robot, we design a tracking and stabilization controller by integrating neural dynamics model into backstepping.

A. *Trajectory Tracking of Mobile Robot*

1) *The Kinematic Controller*

The design of the kinematic controller is based on the kinematic model of the robot described by (1). Equation (8) describes the difference of position and direction of the reference robot from the real robot. The derivative of the trajectory tracking error can be written as

$$\dot{e}_p = \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \\ \dot{e}_\theta \end{bmatrix} = \begin{bmatrix} we_y - v + v_r \cos e_\theta \\ -we_x + v_r \sin e_\theta \\ w_r - w \end{bmatrix} \tag{9}$$

After mentioning the posture tracking error, we have to illustrate the kinematic trajectory tracking control law for the mobile robot. The common velocity command using backstepping method [5] is recommended as follows

$$\begin{cases} v = k_1 e_x + v_r \cos e_\theta \\ w = w_r + k_2 e_y v_r + k_3 \sin e_\theta \end{cases} \tag{10}$$

where k_1, k_2, and k_3 are positive constants and $v_r>0$.

The objective of such a controller is to generate the desired velocities $\vec{v}_c = [v_c, w_c]^T$ for the dynamic controller.

2) *The Dynamic Controller*

The dynamic controller receives from the kinematic controller the desired linear and angular velocities $\vec{v}_c = [v_c, w_c]^T$ which are obtained by (10), and generates another pair of linear and angular velocities to be delivered to robot servos.

We define velocity error $\delta = \vec{v} - \vec{v}_c = [\tilde{v}, \tilde{w}]^T$ and its derivative $\dot{\delta} = \dot{\vec{v}} - \dot{\vec{v}}_c$, then consider the following control law to prepare tracking of v_c and w_c

$$\bar{\tau} = -k_d \delta + \overline{M}\dot{\vec{v}}_c \tag{11}$$

and (11) can be rewritten as

$$\begin{cases} \tau_1 = -k_{d1}\tilde{v} + m\dot{v}_c \\ \tau_2 = -k_{d2}\tilde{w} + I\dot{w}_c \end{cases} \tag{12}$$

where $k_d = diag\,(k_{d1}, k_{d2})$ is a positive matrix.

B. *Point Stabilization of Mobile Robot*

When the robot reaches the virtual target point, the controller should be designed to regulate the robot to real target point. According to Assumption 2 and (10), the kinematic control law which generates desired velocities for dynamic controller can be written as follows

$$\begin{cases} v_c = k_4 e_x \\ w_c = k_5 \sin e_\theta \end{cases} \tag{13}$$

where k_4, k_5 are positive constants.

Similar to dynamic tracking controller, (12) is still suitable to stabilization controller, rewritten as

$$\begin{cases} \tau_1 = -k_{c1}\tilde{v} + m\dot{v}_c \\ \tau_2 = -k_{c2}\tilde{w} + I\dot{w}_c \end{cases} \tag{14}$$

where k_{c1}, k_{c2} are positive constants.

C. *The Proposed Neural Dynamics Based Tracking and Stabilization Controllers*

Analyzing the controller in (10) and (11), we observe that errors e_x, e_y, e_θ, δ and $\dot{\vec{v}}_c$ are not equal to zero at initial status. So the speed jump is caused by the initial tracking errors e_x, e_y and e_θ while the torque jump is produced by velocity error δ and $\dot{\vec{v}}_c$. To solve the two problems for tracking control, e_x, e_y, $\dot{v}_c$ and $\dot{w}_c$ are replaced by s_1, s_2, $\dot{v}_s$ and $\dot{w}_s$, respectively, then a novel tracking controller is proposed as

$$\begin{cases} v_c = k_1 s_1 + v_r \cos e_\theta \\ w_c = w_r + k_2 v_r s_2 + k_3 \sin e_\theta \end{cases} \tag{15}$$

and

$$\begin{cases} \tau_1 = -k_{d1}\tilde{v} + m\dot{v}_s \\ \tau_2 = -k_{d2}\tilde{w} + I\dot{w}_s \end{cases} \tag{16}$$

In (15) and (16), according to (4), s_1, s_2, $\dot{v}_s$ and $\dot{w}_s$ are functions of neural dynamics model given by

$$\begin{aligned} ds_1/dt &= -As_1 + (B - s_1)S^+(e_x) - (D + s_1)S^-(e_x) \\ ds_2/dt &= -As_2 + (B - s_2)S^+(e_y) - (D + s_2)S^-(e_y) \\ d\dot{v}_s/dt &= -A\dot{v}_s + (B - \dot{v}_s)S^+(\dot{v}_c) - (D + \dot{v}_s)S^-(\dot{v}_c) \\ d\dot{w}_s/dt &= -A\dot{w}_s + (B - \dot{w}_s)S^+(\dot{w}_c) - (D + \dot{w}_s)S^-(\dot{w}_c) \end{aligned} \tag{17}$$

where $S^+(x) = \max(0, x)$ and $S^-(x) = \max(0, -x)$.

At initial status, we choose

$$\begin{cases} s_1(0) = -v_r(0)\cos e_\theta(0)/k_1 \\ s_2(0) = -(w_r(0) + k_3 \sin e_\theta(0))/(k_2 v_r(0)) \end{cases} \quad (18)$$

Then, $v_c(0) = 0$ and $w_c(0) = 0$.

If we select following initial values

$$\begin{cases} v(0) = 0 \\ w(0) = 0 \\ \dot{v}_s(0) = 0 \\ \dot{w}_s(0) = 0 \end{cases} \quad (19)$$

According to (16), the control output $\overline{\tau}(0) = [\tau_1(0), \tau_2(0)]^T = [0,0]^T$. The control law (15) (16) is smooth, and whether initial errors exist or not, initial velocities and torques are zero, so the new control law can solve the speed and torque jump problems, respectively.

When the robot is required to stop, the switching between tracking controller and stabilization controller appears. Analyzing the controller in (13) and (14), we find that it is difficult to generate a control signal smoothly because e_x, e_θ, $\dot{v}_c$ and $\dot{w}_c$ are not continuous when controllers switch. Using neural dynamics model, the control law of point stabilization described by (13) and (14) becomes

$$\begin{cases} v_c = k_4 s_1 \\ w_c = k_5 \sin s_3 \end{cases} \quad (20)$$

and

$$\begin{cases} \tau_1 = -k_{c1}\widetilde{v} + m\dot{v}_s \\ \tau_2 = -k_{c2}\widetilde{w} + I\dot{w}_s \end{cases} \quad (21)$$

In (20) and (21), considering (4), s_1, s_3, $\dot{v}_s$ and $\dot{w}_s$ are functions of neural dynamics model given by

$$\begin{aligned} ds_1/dt &= -As_1 + (B - s_1)S^+(e_x) - (D + s_1)S^-(e_x) \\ ds_3/dt &= -As_3 + (B - s_3)S^+(e_\theta) - (D + s_3)S^-(e_\theta) \\ d\dot{v}_s/dt &= -A\dot{v}_s + (B - \dot{v}_s)S^+(\dot{v}_c) - (D + \dot{v}_s)S^-(\dot{v}_c) \\ d\dot{w}_s/dt &= -A\dot{w}_s + (B - \dot{w}_s)S^+(\dot{w}_c) - (D + \dot{w}_s)S^-(\dot{w}_c) \end{aligned} \quad (22)$$

where $S^+(x) = \max(0, x)$ and $S^-(x) = \max(0, -x)$.

To avoid the switching, we make law (20) (21) at P moment equal to rule (15) (16) at (P-1) moment respectively, i.e.

$$\begin{cases} s_1(P) = v_c(P-1)/k_4 \\ s_3(P) = \arcsin(w_c(P-1)/k_5) \\ \dot{v}_s(P) = \dot{v}_c(P-1) \\ \dot{w}_s(P) = \dot{w}_c(P-1) \end{cases} \quad (23)$$

Then, $v_c(P)=v_c(P\text{-}1)$ and $w_c(P)=w_c(P\text{-}1)$, $\tau_1(P)=\tau_1(P\text{-}1)$ and $\tau_2(P)= \tau_2(P\text{-}1)$. The whole control law is continuous and smooth. Hence, the influence of switching between two controllers can be eliminated by using neural dynamics model.

D. Stability analysis

In order to prove the stability of the control system, we consider the following Lyapunov function candidate

$$V = V_1 + V_2 + V_s \quad (24)$$

where

$V_1 = (e_x + e_y)^2/2 + (1 - \cos e_\theta)/k_2$,

$V_2 = \delta^T \overline{M}\delta/2$,

$V_s = (s_1^2 + s_2^2 + s_3^2)/2 + (\dot{v}_s^2 + \dot{w}_s^2)/2$.

herein, s_1, s_2, s_3, $\dot{v}_s$ and $\dot{w}_s$ are defined in (17) and (22). Clearly, $V \geq 0$. Substituting (9)(10) for the time derivative of V_1 and (11) for the time derivative of V_2, we obtain

$\dot{V}_1 = -k_1 e_x^2 - k_3 \sin^2 e_\theta / k_2 \leq 0$,

$\dot{V}_2 = -k_d \delta^T \delta \leq 0$.

Additionally, V_s =0 if and only if s_1, s_2, s_3, $\dot{v}_s$ and $\dot{w}_s$ are zero; otherwise, V_s >0. For the derivative of V_s, we have

$$\begin{aligned} \dot{V}_s &= s_1\dot{s}_1 + s_2\dot{s}_2 + s_3\dot{s}_3 + \dot{v}_s\ddot{v}_s + \dot{w}_s\ddot{w}_s \\ &= -(A + S^+(e_x) + S^-(e_x))s_1^2 \\ &\quad -(A + S^+(e_y) + S^-(e_y))s_2^2 \\ &\quad -(A + S^+(e_\theta) + S^-(e_\theta))s_3^2 \\ &\quad -(A + S^+(\dot{v}_c) + S^-(\dot{v}_c))\dot{v}_s^2 \\ &\quad -(A + S^+(\dot{w}_c) + S^-(\dot{w}_c))\dot{w}_s^2 \end{aligned} \quad (25)$$

According to the definitions (9), if $e_x \geq 0$, $S^+(e_x) = e_x$, and $S^-(e_x) = 0$, then

$$A + S^+(e_x) + S^-(e_x) = A + e_x > 0 \quad (26)$$

If $e_x < 0$, $S^+(e_x) = 0$, and $S^-(e_x) = -e_x$, then

$$A + S^+(e_x) + S^-(e_x) = A - e_x > 0 \quad (27)$$

Similarly, we have that

$A + S^+(e_y) + S^-(e_y) > 0$,

$A + S^+(e_\theta) + S^-(e_\theta) > 0$,

$A + S^+(\dot{v}_c) + S^-(\dot{v}_c) > 0$,

$A + S^+(\dot{w}_c) + S^-(\dot{w}_c) > 0$.

Then the derivative of V_s is always non-positive. Therefore, $\dot{V} \leq 0$, that is to say, the whole system is stable.

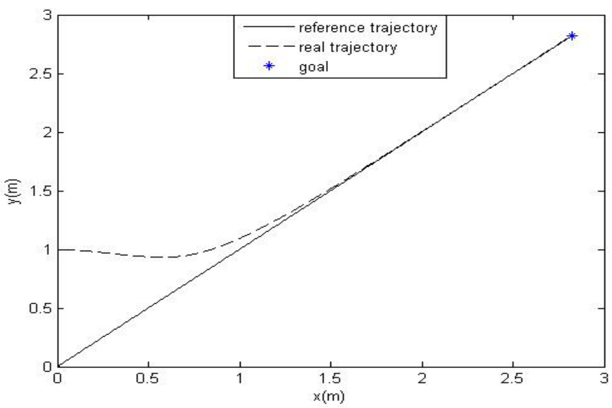
V. SIMULATION RESULTES

In this Section, two test cases in different situations are used to demonstrate the effectiveness of the proposed control scheme, considering the robot's parameters m=30 kg and

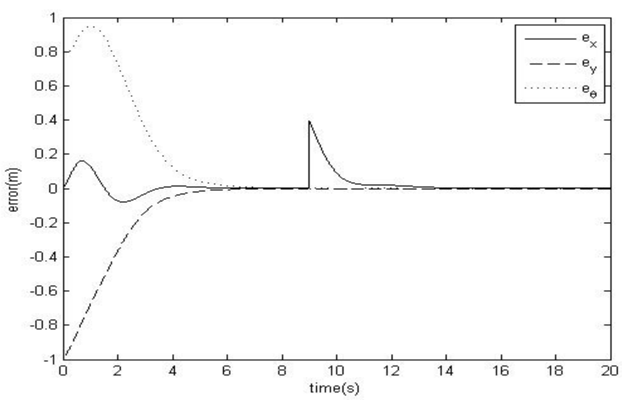
I=15 kg·m^2.The tracking and stabilization performance is shown in Fig. 3 and Fig. 4.

A. Tracking a Straight Line

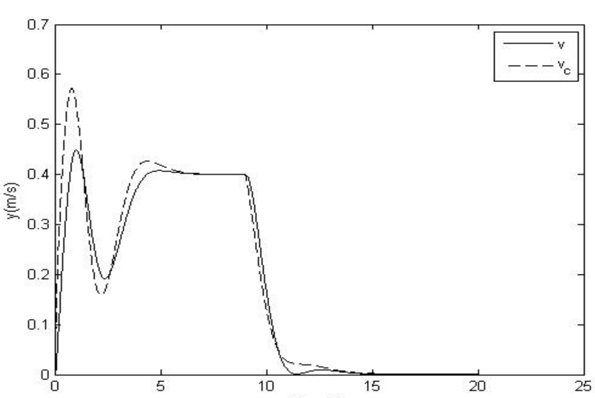
A simple case to track a straight line is studied at first. The straight line trajectory is generated from the reference velocity v_r=0.4 m/s and angular velocity w_r=0 rad/s. The initial posture of the reference trajectory is set at $q_r(0)=[0,0,\pi/4]^T$ while the actual initial posture of robot is $q(0)=[0,1,0]^T$. The target pose calculated by (6) is $q_r(N)=[2.83,2.83,\pi/4]^T$. The tracking results are shown in Fig. 3, including (a) reference and real trajectories, (b) posture errors, (c) actual and desired linear velocities, (d) actual and desired angular velocity, and (e) linear and angular torques.



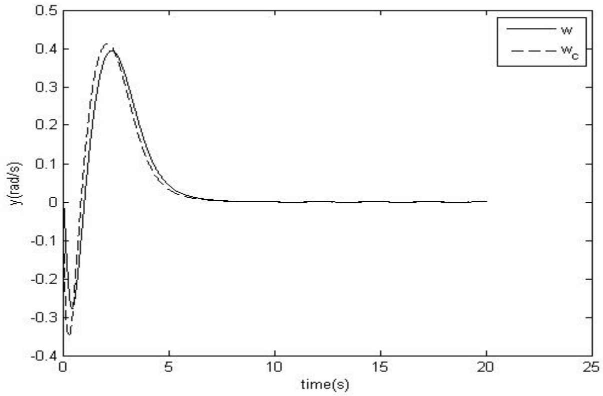
(a). Reference and real trajectories.



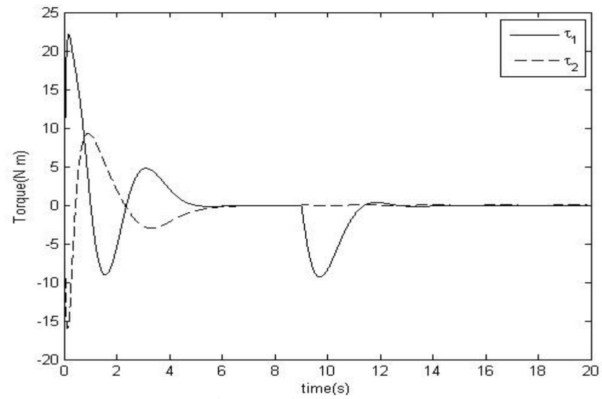
(b). Posture errors.



(c). Actual and desired linear velocities.



(d). Actual and desired angular velocities.



(e). Linear and angular torques.

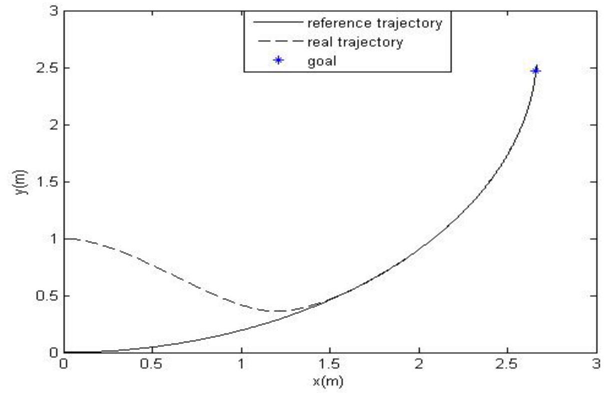
Fig. 3. Tracking results.

From Fig. 3(a), we can see that the robot can track the straight line and stop at the final pose. Fig. 3(b) shows that the controller can correct deviations quickly (about 5.0s) and there are little tracking errors. At t≈9s, the robot is required to park at the goal, and the posture errors in point stabilization are as same as in tracking control except error e_x, so a pulse appears when controllers switch. However, the controller can still regulate the robot to eliminate the error state e_x quickly shown in Fig. 3(e). In addition, the smooth and continuous signals are generated as described in Fig. 3(c, d, e).

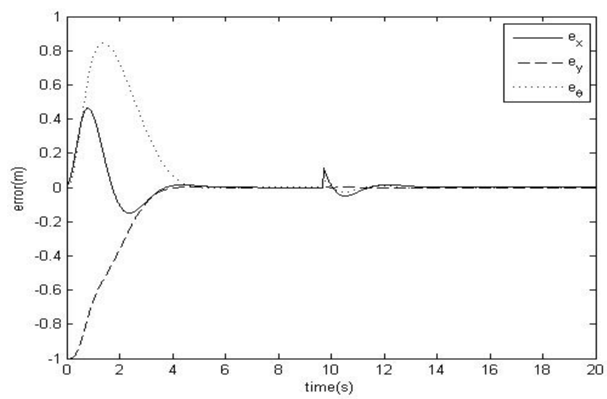
B. Tracking a Curve

The curve trajectory is generated from the reference velocity v_r=0.4 m/s and angular velocity w_r=0.15 rad/s. The initial posture of the reference trajectory is set at $q_r(0)=[0,0,0]^T$ while the actual initial posture is $q(0)=[0,1,0]^T$. The target pose computed by (6) is $q_r(N)=[2.66,2.47,1.50]^T$. The tracking results are shown in Fig. 4.

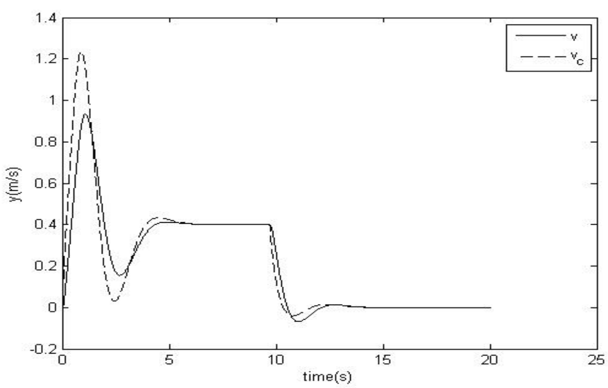
Fig. 4(a) shows that the robot can track the curve and park at the target point. From Fig. 4(b), we can observe that the controller can correct posture errors quickly (about 5.2s) and the steady-state errors can be totally eliminated. Because the initial posture errors in point stabilization exist at t≈10s, small pulses can be seen in Fig. 4(b). But the controller can force the robot to reduce the errors and make them converge to zero quickly. Besides, the real velocities, desired velocities and torques depicted in Fig. 4 (c, d, e) converge to zero, and produce smooth and continuous signals.



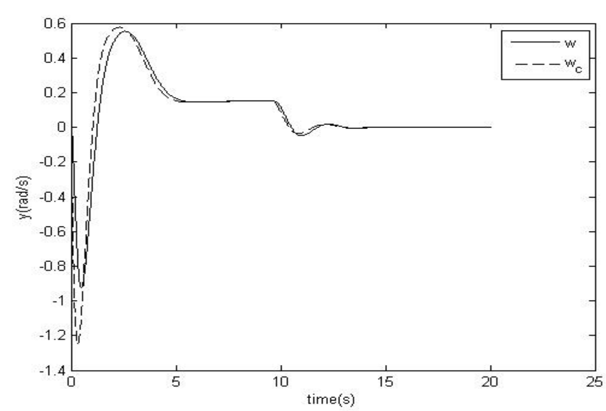
(a). Reference and real trajectories.



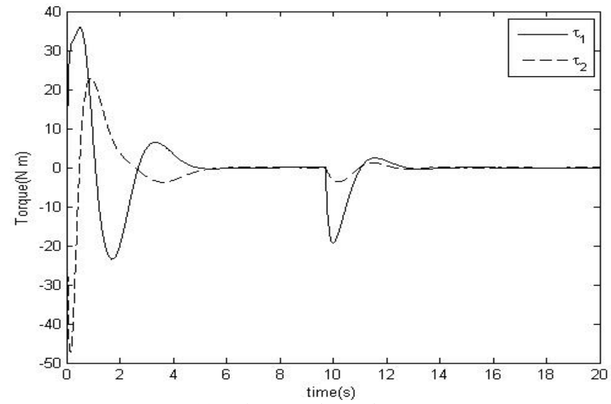
(b). Posture errors.



(c). Actual and desired linear velocities.



(d). Actual and desired angular velocities.



(e). Linear and angular torques.

Fig. 4. Tracking results.

Therefore, all of the simulation results demonstrate that the proposed control strategy is effective to solve the tracking and point stabilization problems.

VI. Conclusion

In this paper, we have proposed a two-stage controller, combined the tracking controller and the point stabilization controller, which allows the mobile robot to track along the reference trajectory and park at the target point. By incorporating a neural dynamics model with the proposed approach, the controller is capable of solving the speed and torque jump problems and generating smooth and continuous control commands. The control scheme is demonstrated to be stable by using Lyapunov theory. All the simulation results indicate that the proposed strategy is indeed feasible and effective.

References

[1] P. A. Nino-Suarez, E. Aranda-Bricaire and M. Velasco-Villa, "Discrete-time sliding mode trajectory-tracking control for a wheeled mobile Robot," in *Proc. 45th IEEE Conf. Decision and Control*, San Diego, 2006, pp. 3025-3057.

[2] S. Chiaverini and G. Fusco, "A fuzzy logic based approach for mobile robot path tracking," *IEEE Transaction on Fuzzy Systems*, vol. 15, no. 2, pp. 211-221, Apr. 2007.

[3] G. Klancar and I. Skrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460-469, Jun. 2007.

[4] X. P. Chen, C. R. Li, G. Y. Li and Y. Y. Luo, "Dynamic model based motor control for wheeled mobile robots," *Robot*, vol. 30, no. 4, pp. 326-332, Jul. 2008.

[5] C. Y. Chen, T. H. Li, Y. C. Yeh and C. C. Chang, "Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots," *Mechatronics*, vol. 19, no.2, pp. 156-166, Mar. 2009.

[6] F. Kuhne, W. F. lages and D. Gomes, "Point stabilization of mobile robots with nonlinear model predictive control," in *IEEE International Conf. Mechatronics and Automation*, Niagara Falls, 2005, pp. 1163-1168.

[7] S. Li, G. L. Ma, and W. L. Hu, "Time-varying adaptive stabilization of an uncertain mobile robot," *Robot*, vol. 27, no. 1, pp. 10-19, Jan. 2005.

[8] H. Y. Yang, S. X. Yang and G. S. Mittal, "Tracking control of a nonholonomic mobile robot by integrating feedback and neural dynamics techniques," in *IEEE International Conf. Intelligent Robots and Systems*, Las Vegas, 2003, pp. 3522-3527.

[9] F. Pourboghrat and M. P. Karlsson, "Adaptive control of dynamic mobile robots with nonholonomic constraints," *Computers and Electrical Engineering*, vol. 28, no. 4, pp. 241-253, Jul. 2002.