Vol 18 No 10, October 2009
1674-1056/2009/18(10)/4222–07

Chinese Physics B

© 2009 Chin. Phys. Soc.
and IOP Publishing Ltd

Collision avoidance for a mobile robot based on radial basis function hybrid force control technique*

Wen Shu-Huan(温淑焕)†

Key Lab of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

(Received 7 December 2008; revised manuscript received 16 April 2009)

Collision avoidance is always difficult in the planning path for a mobile robot. In this paper, the virtual force field between a mobile robot and an obstacle is formed and regulated to maintain a desired distance by hybrid force control algorithm. Since uncertainties from robot dynamics and obstacle degrade the performance of a collision avoidance task, intelligent control is used to compensate for the uncertainties. A radial basis function (RBF) neural network is used to regulate the force field of an accurate distance between a robot and an obstacle in this paper and then simulation studies are conducted to confirm that the proposed algorithm is effective.

Keywords: mobile robot collision avoidance, hybrid force/position control, path planning, RBF neural network

PACC: 0620D, 0630M

1. Introduction

Recently, demand for mobile robot research has been increased enormously due to its characteristic of moving in its terrain. Collision avoidance is always difficult in planning the path for a mobile robot.[1−3] The global path-planning algorithm is based on the environment information that is known. The local path planning algorithm is based on the environment information that is completely unknown or partly unknown. A mobile robot detects the environment around it and receives the information such as position, shape and size of an obstacle. The idea of these algorithms is to draw the map of the environment around the mobile robot through the sensor. With the map, the robot can maintain the desired distance to an obstacle on the path and avoid collision with the obstacle. A mobile robot needs response times to the collision with the obstacle, instead of avoiding collision by using a static map. In this situation, two mobile dynamic systems need a dynamic collision avoidance algorithm. Many collision avoidance algorithms are obtained by the kinematics model of the mobile robot and they ignore the dynamic behaviour.[4−7] The mobile robot dynamic behaviour can improve the performance of the system. It is not adequate to merely use position control to deal with the dynamic interaction position relations. Using force control algorithm is necessary for the dynamic relationship between the mobile robot and the obstacle.[8−10]

In the present paper, a novel collision avoidance algorithm of a mobile robot is proposed. Based on robot dynamics, a collision avoidance algorithm is developed. A virtual force field between a robot and an obstacle is formed and regulated to maintain a desired distance by hybrid force control algorithm. The hybrid force control algorithm is known as one of famous force control algorithms, but uncertainties from incorrect robot dynamic mode degrade the performance of a collision avoidance task. Seul Jung and Eun Soo Jang[11] applied a conventional PID control to a hybrid force control algorithm, it cannot deal with the unexpected. The BP neural network is used to compensate for the robot dynamic uncertainties but the learning time is long.[12] Model-predictive is proposed for collision avoidance of a mobile robot,[13] but it may not suffice for obstacle avoidance in a cramped area. In the present paper, an RBF neural network[14,15] is used to compensate cleverly for the unexpected. As a neural network control algorithm, the reference compensation technique has been used. Simulation studies of avoiding obstacles on the path of a wheeled mobile robot by maintaining a desired distance from the obstacle are conducted to confirm the proposed algorithm.

*Project supported by the Science and Technology Stress Projects of Hebei Province, China (Grant No 07213526).

†E-mail: wenshuhuan@sohu.com

2. Design of system controller

2.1. Kinematics of mobile robot

Figure 1 shows the kinematics coordinate of a mobile robot with the obstacle. The robot is a wheeled drive mobile robot. The point P_f is a virtual contact point where a robot contacts the obstacle. The distance between the centre of a robot and a contact point can be defined as a virtual force. The control algorithm regulates the virtual force. In real applications, this distance can be measured by a distance sensor.

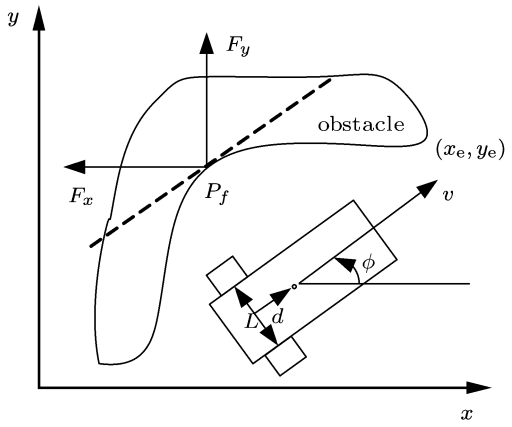


Fig.1. Kinematics of a wheeled robot.

Forward velocity v of a mobile robot and the angular velocity ω at the contact point can be defined by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -d\cdot\sin\phi \\ \sin\phi & -d\cdot\cos\phi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}, \tag{1}$$

where ϕ is the heading angle, d is a distance from a midpoint of a wheel axis to the centre of a robot. Wheel velocities in joint space have the relationship to velocities in Cartesian space as

$$\begin{bmatrix} \nu \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \dot{\theta}_{\mathrm{R}} \\ \dot{\theta}_{\mathrm{L}} \end{bmatrix}, \tag{2}$$

where r is a radius of a wheel, L is the distance between two wheels, $\dot{\theta}_{\mathrm{R}}$ is the velocity of a right wheel, and $\dot{\theta}_{\mathrm{L}}$ is the velocity of left wheel. Combining Eq.(1) with Eq.(2) yields the following Jacobian relationship

$$Z = J\dot{\theta},$$
$$J = \begin{bmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \\ j_{31} & j_{32} \end{bmatrix}, \tag{3}$$

where, $Z = [x \;\; y \;\; \phi]^{\mathrm{T}}$, and each element of the Jacobian matrix is defined as

$$\begin{aligned} j_{11} &= \frac{r}{2}\cos\phi - \frac{r}{L}(d\cdot\sin\phi), \\ j_{12} &= \frac{r}{2}\cos\phi + \frac{r}{L}(d\cdot\sin\phi), \\ j_{21} &= \frac{r}{2}\sin\phi + \frac{r}{L}(d\cdot\cos\phi), \\ j_{22} &= \frac{r}{2}\sin\phi - \frac{r}{L}(d\cdot\cos\phi), \\ j_{31} &= \frac{r}{L}, \\ j_{32} &= -\frac{r}{L}. \end{aligned} \tag{4}$$

2.2. Hybrid force/position control

Hybrid force/position control is known as one of the popular force control algorithms for robot manipulators. This method separates force controlled direction and position controlled direction while impedance force control does not separate the controlled direction.

The wheeled mobile robot dynamic equation can be described as

$$D(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + J^{\mathrm{T}}(\theta)h = \tau, \tag{5}$$

where, $D(\theta) = J^{\mathrm{T}}MJ \in R^{2\times 2}$ and M is the symmetric positive-definite inertia matrix, $C(\theta,\dot{\theta}) = J^{\mathrm{T}}[M\dot{J} + VJ] \in R^{2\times 2}$ and V is the centripetal and Coriolis matrix, $[\theta_R \; \theta_L]^{\mathrm{T}}$ and $h = [f \; \mu]^{\mathrm{T}}$, f is the external force vector, μ is the external moment, τ is the applied torque vector. The wheel velocity can be obtained from the Jacobian relationship

$$\dot{\theta} = J^{-1}\dot{Z}, \tag{6}$$

where Z is the position vector $Z = [x \;\; y \;\; \phi]^{\mathrm{T}}$. Since the inverse of the Jacobian cannot be defined, the pseudo Jacobian matrix can be used.

$$\dot{\theta} = J^{+}\dot{Z}, \tag{7}$$

where $J^{+} = (J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}$. Differentiating expression (7) obtains the acceleration

$$\begin{aligned} \ddot{\theta} &= J^{+}(\ddot{Z} - \dot{J}J^{+}\dot{Z}) \\ &= (J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}(\ddot{Z} - \dot{J}(J^{\mathrm{T}}J)^{-1}J^{\mathrm{T}}\dot{Z}), \end{aligned} \tag{8}$$

where $\dot{Z} = \left[\dot{x}\ \dot{y}\ \dot{\phi}\right]^{\mathrm{T}}$ and $\dot{\theta} = \left[\dot{\theta}_R\ \dot{\theta}_L\right]^{\mathrm{T}}$.
Defining the control law as

$$\tau = \hat{D}(\theta)u + \hat{C}(\theta,\dot{\theta})\dot{\theta} + J^{\mathrm{T}}(\theta)h, \tag{9}$$

where u is the control input vector, $\hat{D}$ and $\hat{C}$ are the estimates of robot model D and C respectively. In the hybrid control, input u can be selectively used by switching algorithm.

The output vector F can be defined as

$$F = [f_x\ f_y\ \phi]^{\mathrm{T}}. \tag{10}$$

Forces in the x axis and the y axis can be defined as

$$f_x = k_{\mathrm{e}}\delta x, \quad f_y = k_{\mathrm{e}}\delta y, \tag{11}$$

where $\delta x = x - x_{\mathrm{e}}$, $\delta y = y - y_{\mathrm{e}}$ and $(x_{\mathrm{e}}, y_{\mathrm{e}})$ is the position of an obstacle. (x, y) is the contact position, k_{e} is the stiffness of an obstacle. Combining expression (10) with expression (11) yields

$$F = K_{\mathrm{e}}\Delta Z = K_{\mathrm{e}}(Z - Z_{\mathrm{e}}) = \begin{bmatrix} k_{\mathrm{e}} & 0 & 0 \\ 0 & k_{\mathrm{e}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_{\mathrm{e}} \\ y - y_{\mathrm{e}} \\ \phi \end{bmatrix}. \tag{12}$$

In general, for simplicity, since $\dot{Z}_{\mathrm{e}} = 0$, $\ddot{Z}_{\mathrm{e}} = 0$, we have

$$Z = K_{\mathrm{e}}^{-1}F. \tag{13}$$

Inserting expression (13) into expression (8) yields

$$\ddot{\theta} = J^{+}(\ddot{Z} - \dot{J}J^{+}\dot{Z}) = J^{+}(K_{\mathrm{e}}^{-1}\ddot{F} - \dot{J}J^{+}K_{\mathrm{e}}^{-1}\dot{F}). \tag{14}$$

From expression (14), substituting $u = \ddot{\theta}$ into expression (9) yields the control law

$$\begin{aligned}\tau = {} & \hat{D}(\theta)[J^{+}(K_{\mathrm{e}}^{-1}\ddot{F} - \dot{J}J^{+}K_{\mathrm{e}}^{-1}\dot{F})] \\ & + \hat{C}(\theta,\dot{\theta})\dot{\theta} + J^{\mathrm{T}}(\theta)h.\end{aligned} \tag{15}$$

In the force controlled direction, F can be formed as a PD type controller

$$u_1 = \ddot{F} = \ddot{F}_d + K_D(\dot{F}_d - \dot{F}) + K_P(F_d - F). \tag{16}$$

Substituting expression (16) into expression (15) yields the control law

$$\begin{aligned}\tau = {} & \hat{D}[J^{+}(K_{\mathrm{e}}^{-1}(\ddot{F}_d + K_D(\dot{F}_d - \dot{F}) + K_P(F_d - F)) \\ & - \dot{J}J^{+}K_{\mathrm{e}}^{-1}\dot{F})] + \hat{C} + J^{\mathrm{T}}h.\end{aligned} \tag{17}$$

In the same way, the control law in the position controlled direction can be obtained as

$$u_2 = \ddot{Z} = \ddot{Z}_d + K_D(\dot{Z}_d - \dot{Z}) + K_P(Z_d - Z). \tag{18}$$

In the position control, since $h = 0$, the control law can be obtained by inserting expression (18) into expression (14), then substituting $u = \ddot{\theta}$ into expression (9) yields

$$\tau = \hat{D}J^{+}[\ddot{Z}_d + K_D(\dot{Z}_d - \dot{Z}) - \dot{J}J^{+}\dot{Z} + K_P(Z_d - Z)] + \hat{C}. \tag{19}$$

In the hybrid force control method, either position control or force control is selectively used by a selection matrix. Selection matrix S can be defined as $S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ for position control and $S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ for force control.

Then overall control law is described as

$$\begin{aligned}\tau = {} & S[\hat{D}J^{+}\{\ddot{Z}_d + K_D(\dot{Z}_d - \dot{Z}) \\ & + K_p(Z_d - Z) - \dot{J}J^{+}Z\} + \hat{C}] \\ & + (I - S)[\hat{D}[J^{+}(K_{\mathrm{e}}^{-1}(\ddot{F}_d + K_D(\dot{F}_d - \dot{F}) \\ & + K_p(F_d - F)) - \dot{J}J^{+}K_{\mathrm{e}}^{-1}\dot{F})] + \hat{C} + J^{\mathrm{T}}h],\end{aligned} \tag{20}$$

where

$$K_{\mathrm{e}} = \begin{bmatrix} k_{\mathrm{e}} & 0 & 0 \\ 0 & k_{\mathrm{e}} & 0 \\ 0 & 0 & k_{\mathrm{e}} \end{bmatrix},$$

$$K_p = \begin{bmatrix} k_{px} & 0 & 0 \\ 0 & k_{py} & 0 \\ 0 & 0 & k_{p\phi} \end{bmatrix},$$

$$K_D = \begin{bmatrix} k_{Dx} & 0 & 0 \\ 0 & k_{Dy} & 0 \\ 0 & 0 & k_{Dy} \end{bmatrix}.$$

When the estimates of dynamic model are exact such as $\hat{D} = D$, $\hat{C} = C$, the closed-loop error equation is

$$\ddot{e} + K_D\dot{e} + K_P e = 0, \tag{21}$$

where $e = F_D - F$, $e = Z_D - Z$.

Figure 2 is the hybrid force/position control block diagram. However, in general, the robot dynamic model cannot be obtained exactly so that equation (21) cannot be achieved. This is the reason why an intelligent method is needed.

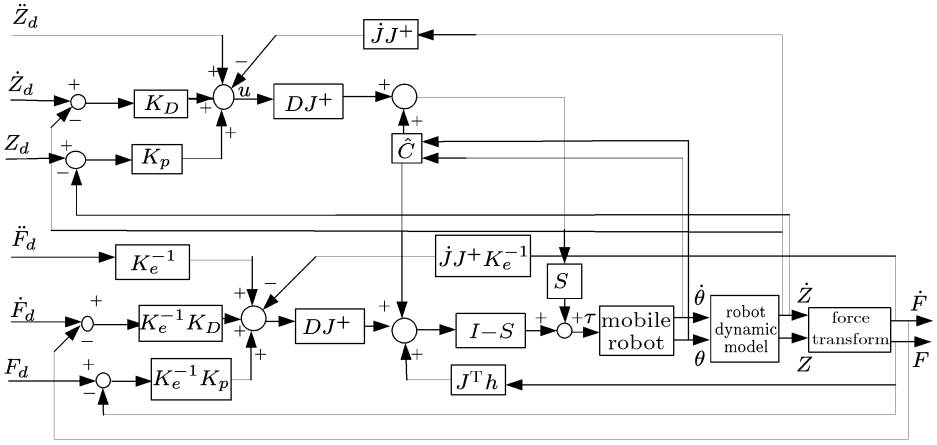


Fig.2. Hybrid force/position control block diagram.

2.3. Intelligent hybrid force/position control based on RBF neural network

Since uncertainties from robot dynamics and obstacles degrade the performance of a collision avoidance task, RBF neural network is used to compensate for uncertainties. RBF neural network compensation signals are added to inputs of PD type controllers to minimize the output errors. In this way, uncertainties can be compensated without intervening internal control structure.

RBF neural network is separately used in position control and force control. Compensation signal ϕ is added to the input level. In the force controlled direction, we have the following control input

$$\ddot{F} = \ddot{F}_d + K_D(\dot{F}_d - \dot{F}) + K_P(F_d - F + \Phi_f), \quad (22)$$

where Φ_f is a compensation signal from RBF neural network.

Inserting Eq.(22) into expression (15) obtains the following control law:

$$\tau = \hat{D}J^+K_{\mathrm{e}}^{-1}[\ddot{F}_d + K_D(\dot{F}_d - \dot{F}) + K_P(F_d - F + \Phi_f) - \dot{J}J^+K_{\mathrm{e}}^{-1}\dot{F}] + \hat{C} + J^{\mathrm{T}}h. \quad (23)$$

In the position controlled direction, the control law is

$$\ddot{Z} = \ddot{Z}_d + K_D(\dot{Z}_d - \dot{Z}) + K_P(Z_d - Z + \Phi_P). \quad (24)$$

Substituting Eq.(24) into expression (15) yields the control law for position control as

$$\tau = \hat{D}J^{\#-1}[Z_d + K_D(\dot{Z}_d - Z) - \dot{J}J^{\#-1}\dot{Z} + K_p(Z_d - Z + \Phi_P)] + \hat{C}. \quad (25)$$

The overall control law can be described as

$$\tau = S\{\hat{D}J^+[\ddot{Z}_d + K_D(\dot{Z}_d - \dot{Z}) - \dot{J}J^+\dot{Z} + K_P(Z_d - Z + \phi_f)] + \hat{C}\} + (1-S)\{\hat{D}J^+K_{\mathrm{e}}^{-1}[\ddot{F}_d + K_D(\dot{F}_d - \dot{F}) + K_P(F_d - F + \phi_P) - \dot{J}J^+K_{\mathrm{e}}^{-1}\dot{F}] + \hat{C} + J^{\mathrm{T}}h\}, \quad (26)$$

where $K_{\mathrm{e}} = \begin{bmatrix} k_{\mathrm{e}} & 0 & 0 \\ 0 & k_{\mathrm{e}} & 0 \\ 0 & 0 & k_{\mathrm{e}} \end{bmatrix}$, $K_p = \begin{bmatrix} k_{px} & 0 & 0 \\ 0 & k_{py} & 0 \\ 0 & 0 & k_{p\phi} \end{bmatrix}$, $K_D = \begin{bmatrix} k_{Dx} & 0 & 0 \\ 0 & k_{Dy} & 0 \\ 0 & 0 & k_{Dy} \end{bmatrix}$.

The intelligent hybrid force/position control block diagram is depicted in Fig.3

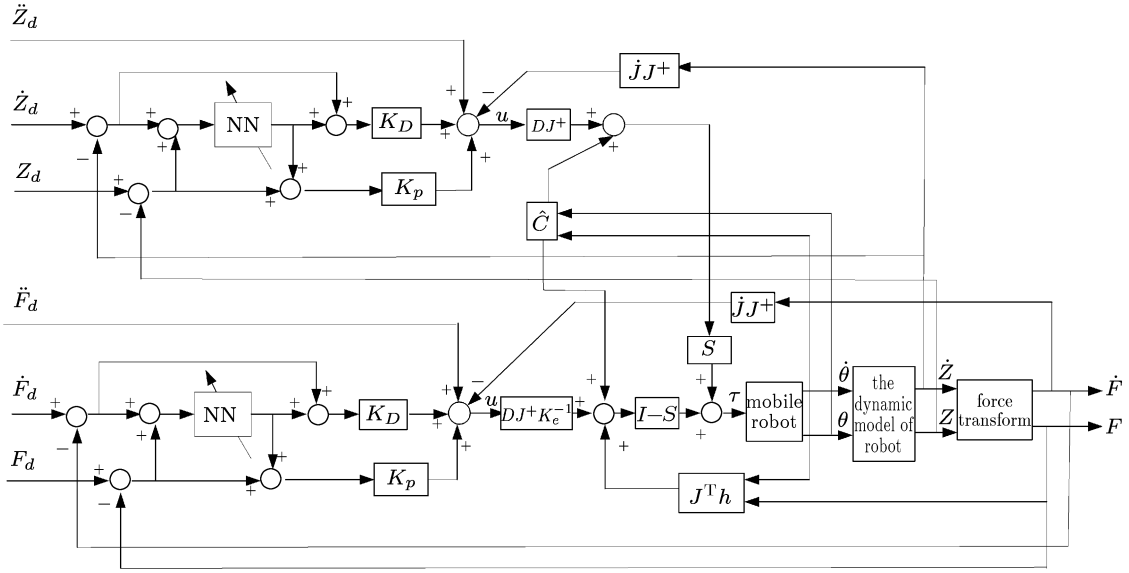


Fig.3. Intelligent hybrid force/position control block diagram.

3. Simulation results and analysis

3.1. Experiment study

Simulation studies are conducted for two cases: one is for conventional PD control and the other is for RBF neural network control. We assume that only 80% of the dynamic model can be estimated, namely $\hat{D} = 0.8D$, $\hat{C} = 0.8C$. The following parameter values of a robot are used in simulations.

Table 1. Parameters of wheeled mobile robot, where m, m_c are the mass of wheeled robot and the mass of non-wheel robot respectively.

parameters	r	L	d	m	m_c
value	0.3 m	0.5 m	0.2 m	0.13 kg	0.5 kg

Moreover, the torque of angular velocities $\dot{\phi}$ in the heading direction is $I_w = 0.2$ Nm. The torque of two wheels is $I = 3$ Nm. The magnitudes of the external force and the external force torque input are as follows: the horizontal component of the external force is $f_1 = 9.2$ N, the vertical component of the external force is $f_2 = 8.7$ N and the external force torque is $\mu = 2$ Nm.

The input signal of position is $Z_d = [x_c \quad y_d \quad \phi_d]^{\mathrm{T}} = [2t, 2t, \pi/4]$, where x_d is the abscissa coordinate, y_d is the vertical coordinate and ϕ_d is the heading angle.

The input signal of force is $F_d = [f_{xd},\ f_{yd},\ f_{id}]^{\mathrm{T}}$, where, $f_{xd} = k_{\mathrm{e}}(x_d - x_{\mathrm{e}})$, $f_{yd} = k_{\mathrm{e}}(y_d - y_{\mathrm{e}})$ are the components of virtual force along the x axis and along the y axis respectivly. f_{id} is the heading angle. Generally, we assume that the coordinate of obstacle (x_{e}, y_{e}) is $x_{\mathrm{e}} = y_{\mathrm{e}}$, $0.4 \le x \le 1$.

The following is to introduce the input and the output signals of RBF neural network:

In the position controlled direction, $e_x = x_d - x$; $e_y = y_d - y$; $e_i = i_d - i$, where x is the distance along the x axis from robot to obstacle, y is the distance along the y axis from robot to obstacle and i is the heading angle of robot. An RBF neural network is proposed to compensate position error.

The inputs of the neural network are e_x, $e_y e_i$, $\dot{e}_x$, $\dot{e}_y$, $\dot{e}_i$ and the output is ϕ_p.

In the force controlled direction, $e_{fx} = f_{xd} - f_x$; $e_{fy} = f_{yd} - f_y$; $e_i = f_{id} - f_i$, where f_x is the components of the virtual force along the x axis, f_y is the components of the virtual force along the y axis and f_i is the heading angle of the robot. The force error is compensated by an RBF neural network whose inputs are e_{fx}, e_{fy}, e_{fi}, $\dot{e}_{fx}$, $\dot{e}_{fy}$, $\dot{e}_{fi}$ and output is ϕ_f.

The following are the simulation results.

3.2. Mobile robot trajectory

Figure 4 shows the trajectory of a mobile robot under an RBF neural network control and under a conventional PD control respectively. It is easily found from Fig.4 that the mobile robot did not collide with the obstacle under both control algorithms, which proves that the hybrid/force control algorithm used in

a collision avoidance for the mobile robot is effective. With the compensation of the RBF neural network, the robot can reach the destination in a shorter path than under the PD control.

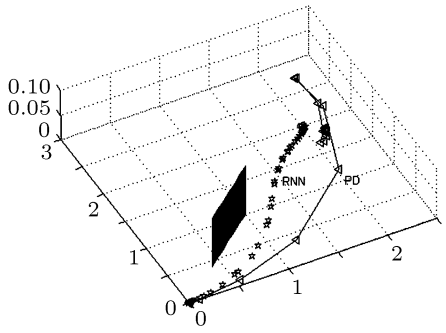


Fig.4. Tracking of mobile robot.

It can be seen from Fig.5 that x component of the mobile robot position error under the RBF neural network control is convergent to zero and the performance under the RBF neural network control is better than under the conventional PD control.

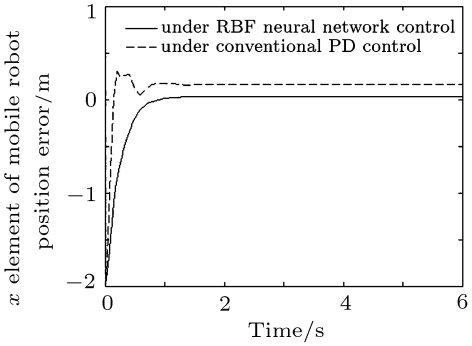


Fig.5. The x component of mobile robot position error.

Figure 6 demonstrates that y components of the mobile robot position error under the RBF neural network control is distinctly smaller than under the conventional PD control. It can be seen from Fig.6 that

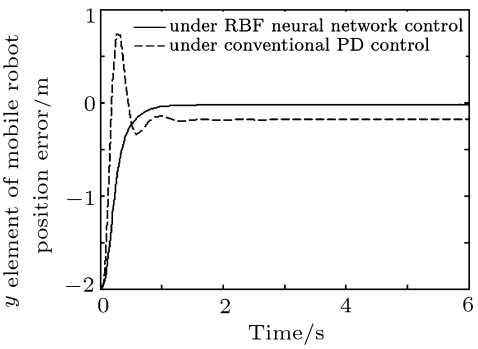


Fig.6. The y components of mobile robot position error.

the performance of the mobile robot nearly reaches the desired value.

As shown in Fig.7, the regulation time is much shorter and the overshoot is less under the RBF neural network control.

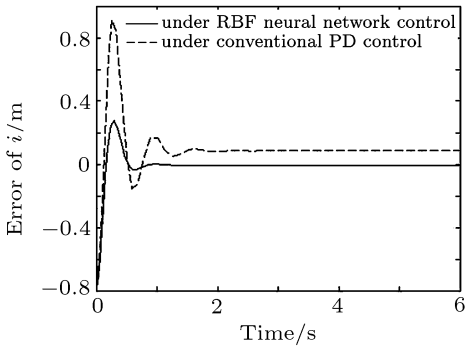


Fig.7. Error of i.

The simulation results of the trajectory of the mobile robot with respect to the task coordinate system are shown in these figures above. These figures show that the intelligent path planning and precise control are realized in the system under the RBF neural network control and the robot reaches the destination accurately and takes less time.

3.3. Rotation angle of mobile robot wheel

Figures 8 and 9 show a comparison between the heading angles of a mobile robot. The outlines of the figures are similar. The fluctuation is smaller and the regulation time is shorter under the RBF neural network. It is seen from these figures that the control efficiencies are considerably improved.

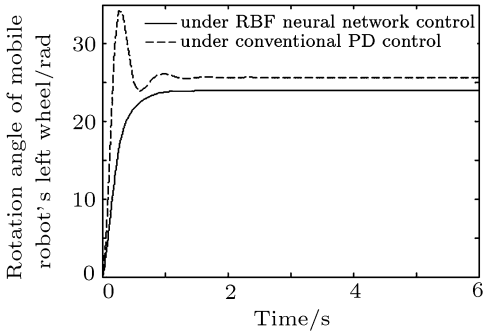


Fig.8. Rotation angle of left wheel of mobile robot.

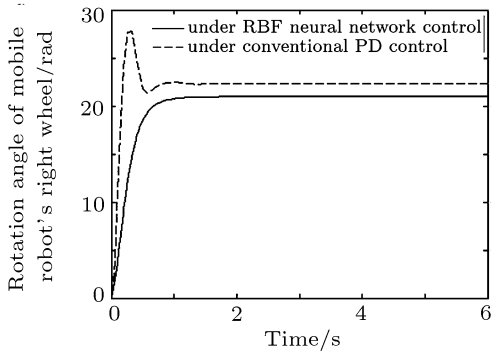


Fig.9. Rotation angle of right wheel of mobile robot.

In order to improve the capability of dynamic obstacle avoidance for wheeled mobile robots, a method of dynamic obstacle avoidance in which the hybrid force/position control is used is presented. Because of uncertainties of the environment and the mobile robot, it is difficult to achieve a good control performance by using the conventional PD controller. So the RBF neural network controller is proposed in this paper. Simulation results show the effectiveness of the RBF neural network control.

4. Conclusions

In this paper, a novel intelligent hybrid force/position control technique is proposed for collision avoidance of a mobile robot. The virtual force field between a robot and an obstacle is formed and regulated to maintain a desired distance by the hybrid force control algorithm. An RBF neural network control strategy is used in hybrid force/position control to compensate for the uncertainties. Compared with conventional PD control, the system with uncertainties under the RBF neural network has a better real-time performance. The robot can find the shorter path without collision with the obstacle and it takes less time to arrive at the destination. The position tracking error is shown to converge to zero. All the computer simulations demonstrate that the collision avoidance performance with using the RBF neural network control algrithm for mobile robot is indeed feasible and effective.

References

[1] Duan Y and Xue X H 2006 *Control Theory & Applications* **23** 1009

[2] Sabry Hassouna M, Abdel-Hakim A E and Farag A A 2009 *Image and Vision Computing* **27** 10

[3] Du J J and Xi Y G 2006 *Control Engineering of China* **13** 183

[4] Sankaranarayanan V and Mahindrakar A D 2009 *Communications in Nonlinear Science and Numerical Simulation* **14** 2319

[5] Wang Z M, Yue H, Li C G and Liu J Y 2006 *Mechanical Science and Technology* **25** 21

[6] Wang Z M, Yue H and Liu J Y 2005 *Journal of Taiyuan University of Technology* **36** 474 (in Chinese)

[7] Takateru U, Kenji O and Yukio T 2004 *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS) **3** 2428

[8] Cai Z X 2000 *Robotics* (Beijing: Tsinghua University Press) p153 (in Chinese)

[9] Jung S, Lasky T A and Hsia T C 2005 *IEEE International Conferenceon Advanced Intelligent Mechatronics* **1** 265

[10] Yu A L 2008 *Acta Phys. Sin.* **57** 3385 (in Chinese)

[11] Seul J and Eun S J 2004 *IECON Proceedings (Industrial Electronics Conference)* **1** 413

[12] Seul J, Eun S J and Hsia T C 2005 *Intelligent Robots and Systems* **2** 382

[13] Yoon Y, Shin J and Kimb H J 2009 *Control Engineering Practice* **15** 1

[14] Zhang J F and Hu S S 2007 *Acta Phys. Sin.* **56** 713 (in Chinese)

[15] Guo H J and Liu R H 2004 *Acta Phsy. Sin.* **53** 4080 (in Chinese)