Springer

Journal of Mechanical Science and Technology

Journal of Mechanical Science and Technology 22 (2008) 2403~2416

www.springerlink.com/content/1738-494x
DOI 10.1007/s12206-008-0746-5

Receding horizon tracking control for wheeled mobile robots with time-delay †

Yu Gao, Chang Goo Lee and Kil To Chong*

Institute of Information and Communication, Chonbuk National University, Jeonju, 561-765, Korea

(Manuscript Received January 16, 2008; Revised May 26, 2008; Accepted July 26, 2008)

Abstract

In this paper, a receding horizon (RH) controller is developed for tracking control of wheeled mobile robots (WMRs) subject to nonholonomic constraint in the environments without obstacles. The problem is simplified by neglecting the vehicle dynamics and considering only the steering system. First, the tracking-error kinematic model is linearized at the equilibrium point. And then, it is transferred to an exact discrete form considering the time-delay. The control policy is derived from the optimization of a quadratic cost function, which penalizes the tracking error and control variables in each sampling time. The minimizing problem is solved by using the QP (quadratic programming) method taking the current error state as the initial value and including the velocity constraints. The performance of the control algorithm is verified via the computer simulations with several different predefined trajectories showing that the strategy is feasible.

Keywords: Receding horizon control; Trajectory tracking; Wheeled mobile robot; Quadratic programming

1. Introduction

In recent years, there has been an increasing amount of research on the subject of mobile robots, which are widely used in many areas. As the most popular kind, the differentially steered wheeled mobile robots (WMRs) have high mobility, high traction with pneumatic tires, and a simple wheel configuration. There are several kinds of controllers proposed for mobile robots with nonholonomic constraints, for which two main approaches to control mobile robots are point stabilization and trajectory tracking.

The aim of point stabilization can be regarded as the generation of control inputs to drive the robot from an initial point to a target point, such as in [1, 2, 3]. Differently, trajectory tracking is to have the robot moving follow a reference trajectory, which is easier to achieve than point stabilization and more natural for a mobile robot. Usually, the reference trajectory can be obtained by a reference robot, and all the kinematic constraints are implicitly considered by the reference trajectory. Some early research by Song and Li [4] developed an LQR controller based on a linearized state-space model. In their presentation, the tracking errors can be eliminated and the mobile robot can follow the specified trajectories. In the linear model approach, however, the controller works only when the linear velocity is not zero. Under such circumstances, it would be difficult to control the mobile robot to track the specified trajectory and in the meantime stop with the specified pose. Consequently, a more generalized approach is desirable. Nonlinear system theory has been employed to solve this problem, such as [5, 6]. Two main research directions employing nonlinear control design can be distinguished. The first uses discontinuous feedback, whereas the second research direction uses time-varying continuous feedback. However, though these solve the regulation problem, they yield slow asymptotic convergence. To obtain faster convergence (e.g.,

† This paper was recommended for publication in revised form by Associate Editor Doo Yong Lee

*Corresponding author. Tel.: +82 63 270 2478, Fax.: +82 63 270 2457
E-mail address: kitchong@chonbuk.ac.kr

exponential convergence), an alternative approach was initially proposed by M'Closkey and Murray [7], and research on the tracking problem for mobile robots has been extensive. With the input saturations, global solutions to the stabilization and tracking problem for the kinematic model were derived and a time-varying state-feedback controller was obtained in [8, 9]. Input-output linearization is used in [10] for nonlinear systems having more outputs than inputs, by using a generalized inverse concept. [11] presented an output-feedback controller that forced the output (position and orientation) of a unicycle-type mobile robot to track a predefined path. A coordinate transformation was first derived to cancel the velocity quadratic terms. An observer was then designed to globally exponentially/asymptotically estimate the unmearsured velocities. And more research can be seen in [12-14].

Receding horizon control, also called model predictive control, is a technology in widespread use in industry for control design of highly complex multivariable processes [15-19]. Such an RHC method represents a way of transforming an open-loop design methodology (i.e., optimal control) into a feedback one, as at every time step the input applied to the process depends on the most recent measurements. In the field of mobile robotics predictive approaches to path tracking also seem to be very promising because the reference trajectory is known beforehand [20]. showed how to improve the robustness of mobile robot path tracking when predictive control algorithms were used, and [21] presented a new path-tracking scheme for a car-like mobile robot based on neural predictive control.

This paper deals with differentially steered wheeled mobile robots and RH trajectory-tracking control with time-delay on a reference trajectory. The problem is simplified by neglecting the vehicle dynamics and considered only the steering system. To compute the vehicle control inputs, it is assumed that there is "perfect velocity tracking" [22]. The control law is based on an error kinematic model, which is linearized at the equilibrium point. This model has stabilization by using linear feedback for the case that direct and angle velocities are persistent. RHC is a one-step-ahead predictive controller obtained by minimizing the difference between the future trajectory-following errors of the robot and the reference point. In time-free systems, the control law can be obtained by minimizing the quadratic cost function consisting of tracking errors and control effort. The QP (quadratic programming) method is used to solve such optimal problem with the constraints. But in time-delay systems, first, the exact discrete-time error-tracking model is obtained by a short sampling time. Then, the QP method is applied to solve the tracking problem with delay. Due to the more complex control structure and taking into account future tracking-errors of the robot, the RHC gives good tracking results. However, compared to other related control algorithms, the main advantage of the approach is the lower computation burden.

A preliminary study on the control of wheeled mobile robot is presented in this paper. It is organized into six sections to accurately present our approach. The dynamics and kinematics of the mobile robot are introduced in Section 2. Section 3 discusses the construction of the error-tracking control. The receding horizon control is first introduced in the Section 4. The RH controller is designed for the error-tracking model with time-delay and the problem is solved by the QP method. The computer simulation results of different trajectories are shown in Section 5. Finally, the conclusions of our study are drawn.

2. Dynamics and kinematics of the wheeled mobile robot

A mobile robot system with n generalized coordinates $(q_1,\ldots,q_n)$ and subject to m constraints can be described by [23, 24].

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = B(q)\tau - A^T(q)\lambda \tag{1}$$

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, $G(q) \in \mathbb{R}^{n\times 1}$ is the gravitational vector. $B(q) \in \mathbb{R}^{n\times r}$ is the input transformation matrix, $\tau \in \mathbb{R}^{r\times 1}$ 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.

All kinematic equality constraints are independent of time and can be expressed as follows:

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

Let $S(q)$ be a full rank matrix $(n-m)$ formed by a set of smooth (i.e., continuously differentiable) and linearly independent vector fields in the null space of $A(q)$:

$$S^T(q)A^T(q) = 0 \tag{3}$$

According to (2) and (3), it is possible to find an auxiliary vector time function $u(t) \in \mathbb{R}^{n-m}$ such that, for all t

$$\dot{q} = S(q)u(t) \tag{4}$$

The mobile robot shown in Fig. 1 is a typical example of a nonholonomic mechanical system. It consists of a vehicle with two driving wheels mounted on the same axis and two castors. The motion and orientation are achieved by independent actuators, e.g., dc motors providing the necessary torques to the rear wheels.

The position and orientation of the robot in an interial Cartesian frame $\{O, X, Y\}$ are completely specified by $q = [x \quad y \quad \theta]^T$, which are the coordinates of P_c. P_c is the center of the axis of the driving wheels and we also assume it as the mass center of the robot in our paper.

The nonholonomic constraint states that the robot can only move in the direction normal to the axis of the driving wheels, i.e., the robot satisfies the condition of nonslipping in a lateral direction:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{5}$$

It is easy to verify that $S(q)$ is given by

$$S(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \tag{6}$$

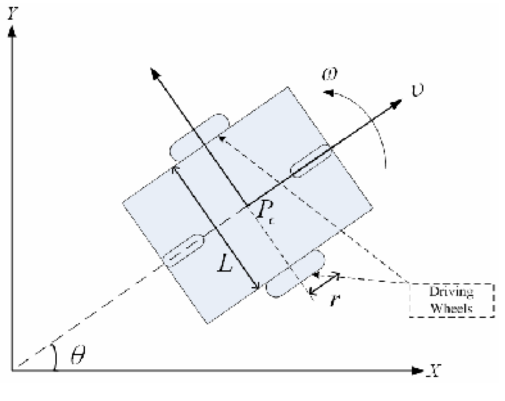


Fig. 1. A nonholonomic wheeled mobile robot.

The kinematic equations of motion (4) in terms of its linear speed υ and angular speed ω are

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} u, \quad u = \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} \tag{7}$$

which is called the steering system of the vehicle.

The dynamic equations of the mobile robot base in Fig. 1 can be expressed as follows:

$$M(q) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}, V_m(q,\dot{q}) = 0, G(q) = 0,$$

$$B(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ \frac{L}{2} & -\frac{L}{2} \end{bmatrix}, \tau = \begin{bmatrix} \tau_r \\ \tau_l \end{bmatrix}, A^T(q) = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix}$$

$$\lambda = -m(\dot{x}\cos\theta + \dot{y}\sin\theta)\dot{\theta} \tag{8}$$

where m is the mass of the robot, I is the moment of inertia of the robot about a vertical axis through P_c, τ_r and τ_l are the torques acting on the wheel generated by the right and left motors.

The system (1) is now transformed into a more appropriate representation for controls purposes. Differentiating (4), substituting this result in (1), and then multiplying by S^T, we can eliminate the constraint matrix $A^T(q)\lambda$. The complete equations of motion of the nonholonomic mobile platform are given by

$$\dot{q} = Su \tag{9}$$

$$(S^T MS)\dot{u} + (S^T M\dot{S})u + \overline{F} = S^T B\tau \tag{10}$$

It is desirable to select a velocity control $u(t)$ for the tracking task only in the steering system (9) by ignoring the dynamics (10) temporarily.

3. Definition of error-tracking control

3.1 Error-tracking model

The reference trajectories should also be described by a reference state vector $q_r = (x_r, y_r, \theta_r)^T$ and a reference control signal vectors $u_r = (\upsilon_r, \omega_r)^T$, which have the same kinematic Eq. (7):

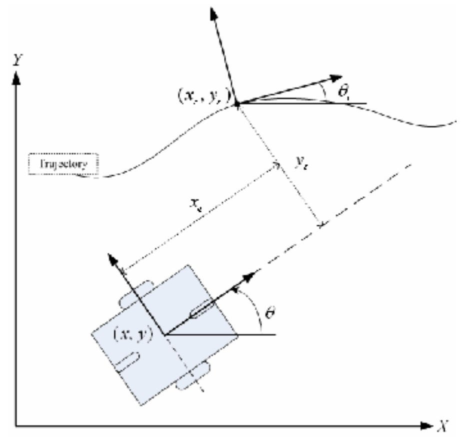


Fig. 2. Robot following error transformation.

$$\dot{q}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \upsilon_r \cos\theta_r \\ \upsilon_r \sin\theta_r \\ \omega_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} u_r \tag{11}$$

To control (7) and to track (11), an error state can be expressed in the frame of the robot (see [22]), as shown in Fig. 2:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_r - q) \tag{12}$$

Therefore, the tracking error model is obtained (see the Appendix. A):

$$\begin{aligned} \dot{x}_e &= \omega y_e - \upsilon + \upsilon_r \cos\theta_e \\ \dot{y}_e &= -\omega x_e + \upsilon_r \sin\theta_e \\ \dot{\theta}_e &= \omega_r - \omega \end{aligned} \tag{13}$$

3.2 Control construction

Refining the control signals

$$u_e = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} \upsilon - \upsilon_r \cos\theta_e \\ \omega - \omega_r \end{bmatrix} \tag{14}$$

Rewriting the Eq. (13) results in the following tracking-error model:

$$\dot{e} = \begin{bmatrix} 0 & \omega & 0 \\ -\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} \upsilon_r + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_e \tag{15}$$

In the Eq. (7), we know that the input of the kinematic equation is $u = (\upsilon, \omega)^T$. Now, the input vector u can be defined again, as the sum of the feedforward and feedback control signals, as follows:

$$u = u_{forward} + u_{back} \tag{16}$$

where the feedforward input vector

$$u_{forward} = \begin{bmatrix} \upsilon_r \cos e_3 \\ \omega_r \end{bmatrix} \tag{17}$$

is obtained by a nonlinear transformation of the velocities of the reference point. And the feedback input (i.e., the error control signal)

$$u_{back} = u_e = \begin{bmatrix} u_{e1} \\ u_{e2} \end{bmatrix} \tag{18}$$

which is used in Eq. (15).

The feedforward control law is derived from a given reference trajectory $(x_r(t), y_r(t))$ defined in a time interval $t \in [0, t]$. However, the calculated robot inputs drive the robot on a desired path only if there are no disturbances and no initial state errors. The nominal tangential $\upsilon_r(t)$ is calculated as follows:

$$\upsilon_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{19}$$

where the sign depends on the desired drive direction (i.e., + for forward and − for reverse). The tangent angle of each point on the path is defined as

$$\theta_r(t) = \mathrm{ATAN2}\left(\dot{y}_r(t), \dot{x}_r(t)\right) + k\pi,\ k = 0,1 \tag{20}$$

where ATAN2 is the four-quadratic inverse tangent function (undefined only if both arguments are zero). By calculating the time derivative of Eq. (20), the nominal angular velocity $\omega_r(t)$ is obtained:

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \dot{y}_r(t)\ddot{x}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{21}$$

By using relations Eq. (19), Eq. (21) and the defined reference robot path $q_r = (x_r, y_r, \theta_r)^T$, the nominal inputs $\upsilon_r(t)$ and $\omega_r(t)$ of the reference point are calculated. In order to be exactly reproducible using $\upsilon_r(t)$ and $\omega_r(t)$, the desired Cartesian motion ($x_r(t), y_r(t)$) should be twice differentiable.

The feedback input is derived in Eq. (16) and Eq. (18). Subsequently, by linearizing the error model (15) at the equilibrium point $(e_1 = e_2 = e_3 = 0,$ $u_{e1} = u_{e2} = 0)$, the following linear model is obtained (see the Appendix. B):

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & \upsilon_r \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_e \tag{22}$$

Eq. (22) is in the state-space form, $\dot{e} = A_c e + B_c u_e$. The controllability matrix $[B_c \;\; A_c B_c \;\; A_c^2 B_c]$ has full rank if υ_r or ω_r is nonzero, which is a sufficient condition for controllability only when the reference inputs υ_r and ω_r are constant (linear and circular paths). Since the error state model (22) is controllable, a local asymptotic stable controller can be found (see [22]), by using the "persistent excitation" of $\omega_r(t)$ and $\upsilon_r(t)$ ($\lim_{t\to\infty} \omega_r^2(t) + \upsilon_r^2(t) \neq 0$). Now we have obtained the differential equation form.

4. Receding horizon tracking controller

Receding horizon control (RHC) is a widely used technology in industry for control design of highly complex multivariable processes. The idea behind RHC is to start with a model of the open-loop process that explains the relations among the system variables (command inputs, internal states, and measured outputs). Then, constraint specifications on system variables are added, such as input limitations and desired ranges where states and outputs should remain. Desired performance specifications complete the control problem setup and are expressed through different weights on tracking errors and actuator efforts. The rest of the RHC design is automatic. First, an optimal control problem based on the given model, constraints, and weights, is constructed and translated into an equivalent optimization problem, which depends on the initial state and reference signals. Then, at each sampling time, the optimization problem is solved by taking the current (measured or estimated) state as the initial state of the optimal control problem. For this reason the approach is said to be predictive, as in fact the optimal control problem is formulated over a time-interval that starts at the current time up to a certain interval in the future. The result of the optimization is an optimal sequence of future control moves. Only the first sample of such a sequence is actually applied to the process; the remaining moves are discarded. At the next time step, a new optimal control problem based on the new feedback state is solved over a shifted prediction horizon. For this reason the approach is also called "receding horizon" control.

4.1 Problem formulation

The idea of the receding horizon control concept is to find the control variable values that minimize the quadratic cost function based on the predicted robot-following error:

$$V(\varepsilon, u_e) \triangleq \sum_{j=1}^{h} \left[\varepsilon^T(k+j) Q \varepsilon(k+j) + u_e^T(k+j-1) R u_e(k+j-1) \right] \tag{23}$$

where $\varepsilon(k+j) = e_r(k+j) - e(k+j)$. $e_r(k+j)$ and $e(k+j)$ stand for the reference robot-tracking-error and real robot-tracking-error, respectively; h is the prediction horizon and Q, R are weighting matrices for error state and control variables where $Q \in \mathbb{R}^n \times \mathbb{R}^n$ and $R \in \mathbb{R}^m \times \mathbb{R}^m$.

4.2 Exact discrete-time error-tracking model with time-delay

Considering the time-delay, the linear error-tracking model (22) becomes

$$\dot{e} = A_c e + B_c u_e(t - D) \tag{24}$$

Set the sampling time interval T_s and assume that delay time $D = (d-1)T_s + \gamma$. We can verify that the delayed input variable attains the following two distinct values within the sampling interval (shown in Fig. 3):

$$\begin{aligned} &\textit{if} \quad kT_s \le t < kT_s + \gamma \\ &\qquad u(t-D) = u(k-d) \\ &\textit{if} \quad kT_s + \gamma \le t < (k+1)T_s \\ &\qquad u(t-D) = u(k-d+1) \end{aligned} \tag{25}$$

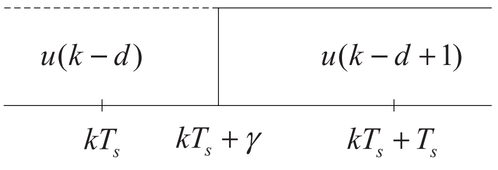


Fig. 3. Scheme of input with time-delay.

So the error-tracking model can be written in the exact discrete-time form as

$$e(k+1) = \Phi e(k) + \Gamma u_e(k-d) + \bar{\Gamma} u_e(k-d+1) \quad (26)$$

where

$$\Phi = e^{A_c T_s} \quad (27)$$

$$\Gamma = \int_0^{\gamma} e^{A_c \tau} d\tau \quad (28)$$

$$\bar{\Gamma} = \int_0^{T_s-\gamma} e^{A_c \tau} d\tau \quad (29)$$

4.3 Prediction in the discrete-time framework with time-delay

In the moving time frame, the predictive error state within the horizon prediction h can be written as:

$$\begin{aligned}
e(k+1) &= \Phi(k)e(k) + \Gamma(k)u_e(k-d) + \bar{\Gamma}(k)u_e(k-d+1),\\
e(k+2) &= \Phi(k+1)\Phi(k)e(k) + \Phi(k+1)\Gamma(k)u_e(k-d)\\
&\quad + [\Phi(k+1)\bar{\Gamma}(k) + \Gamma(k+1)]u_e(k-d+1)\\
&\quad + \bar{\Gamma}(k+1)u_e(k-d+2),\\
&\vdots\\
e(k+h) &= \prod_{j=0}^{h-1}\Phi(k+j)e(k) + \prod_{j=1}^{h-1}\Phi(k+j)\Gamma(k)u_e(k-d)\\
&\quad + \left[\prod_{j=2}^{h-1}\Phi(k+j)\Gamma(k+1) + \prod_{j=1}^{h-1}\Phi(k+j)\bar{\Gamma}(k)\right]u_e(k-d+1)\\
&\quad \cdots + \left[\Gamma(k+h-1) + \Phi(k+h-1)\bar{\Gamma}(k+h-2)\right]u_e(k-d+h),
\end{aligned} \quad (30)$$

Now it is possible to recast the optimization problem in the usual quadratic programming form. Hence, we introduce the following vectors:

$$E(k) \triangleq \begin{bmatrix} e(k+1) \\ e(k+2) \\ \vdots \\ e(k+h) \end{bmatrix}, \quad U_e(k) \triangleq \begin{bmatrix} u_e(k-d) \\ u_e(k-d+1) \\ \vdots \\ u_e(k-d+h) \end{bmatrix} \quad (31)$$

where $E(k) \in \mathbb{R}^{n\cdot h}$ and $U_e(k) \in \mathbb{R}^{m\cdot(h+1)}$.

The predictive error state can be rewritten in the form

$$E(k) = F(k)e(k) + G(k)U_e(k) \quad (32)$$

where

$$F(k) = \begin{bmatrix} \Phi(k) \\ \Phi(k+1)A(k) \\ \vdots \\ \prod_{j=0}^{h-1}\Phi(k+j) \end{bmatrix} \quad (33)$$

and

$$G(k) = [G_1(k) \quad G_2(k) \quad G_3(k) \quad \dots \quad G_{h+1}(k)],$$
$$G(k) \in \mathbb{R}^{n\cdot h} \times \mathbb{R}^{m\cdot(h+1)} \quad (34)$$
$$G_1(k) = \begin{bmatrix} \Gamma(k) \\ \Phi(k+1)\Gamma(k) \\ \vdots \\ \Lambda(k,1)\Gamma(k) \end{bmatrix},$$
$$G_2(k) = \begin{bmatrix} \bar{\Gamma}(k) \\ \Phi(k+1)\bar{\Gamma}(k) + \Gamma(k+1) \\ \vdots \\ \Lambda(k,2)\Gamma(k+1) + \Lambda(k,1)\bar{\Gamma}(k) \end{bmatrix},$$
$$G_3(k) = \begin{bmatrix} 0 \\ \bar{\Gamma}(k+1) \\ \vdots \\ \Lambda(k,3)\Gamma(k+2) + \Lambda(k,2)\bar{\Gamma}(k+1) \end{bmatrix},$$
$$G_{h+1}(k) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Gamma(k+h-1) + \Lambda(k,h-1)\bar{\Gamma}(k+h-2) \end{bmatrix},$$
$$\Lambda(k,i) = \left(\prod_{j=i}^{h-1}\Phi(k+j)\right)$$

The objective of the control law is to drive the predictive robot trajectory as close as possible to the future reference trajectory. This implies that the future reference signal needs to be known. Let us define the reference error-tracking trajectory in state-space as

$$e_r(k+j) = A_r^j e(k) \quad (35)$$

for $j = 1,2,...,h$. This means that the future control error should decrease according to the reference model matrix $A_r \triangleq diag(\delta_1,...,\delta_n), A_r \in \mathbb{R}^n \times \mathbb{R}^n$, where $0 < \delta_{1,2,...,n} < 1$, is the decreasing parameter corresponding to each state [18].

Defining the robot reference-tracking-error vector

$$E_r(k) = \begin{bmatrix} e_r(k+1) \\ e_r(k+2) \\ \vdots \\ e_r(k+h) \end{bmatrix} \tag{36}$$

where $E_r(k) \in \mathbb{R}^{n \cdot h}$. It can be rewritten as follows:

$$E_r(k) = F_r e(k) \tag{37}$$

where

$$F_r = \begin{bmatrix} A_r \\ A_r^2 \\ \vdots \\ A_r^h \end{bmatrix} \tag{38}$$

and $F_r \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$.

4.4 Quadratic programming and control law

The idea of RHC is to minimize the difference between the predictive robot tracking-error and the reference tracking-error in a certain receding horizon interval.

The cost function can be written as follows:

$$\begin{aligned} V(E(k), U_e(k)) = (E_r(k) - E(k))^T \bar{Q}(E_r(k) \\ - E(k)) + U_e(k)^T \bar{R} U_e(k) \end{aligned} \tag{39}$$

where $\bar{Q} \triangleq diag\{Q, \ldots Q\}$, $\bar{Q} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{n \cdot h}$. And $\bar{R} \triangleq diag\{R, \ldots R\}$, $\bar{R} \in \mathbb{R}^{m \cdot (h+1)} \times \mathbb{R}^{m \cdot (h+1)}$.

Putting Eq. (32) and Eq. (36) into the above equation

$$V(U_e(k)) = \bar{V} + \frac{1}{2} U_e(k)^T \cdot H \cdot U_e(k) + f \cdot U_e(k) \tag{40}$$

where

$$\begin{aligned} &\bar{V} = e(k)^T [F_r - F(k)]^T \bar{Q} [F_r - F(k)] e(k) \\ &H = 2(G(k)^T \bar{Q} G(k) + \bar{R}) \\ &f = -2[G(k)^T \bar{Q}(F_r - F(k)) e(k)]^T \end{aligned} \tag{41}$$

Eq. (40) becomes a standard form of the QP problem [25]. The matrix H is a Hessian matrix that is always positive definite. It describes the quadratic part of the cost function and the vector f describes the linear part. $\bar{V}$ is independent of $U_e(k)$ and has no influence in the determination of the control law.

The control consequence can be obtained from the above equation as follows:

$$U_e^{OPT}(k) = arc\min(V(U_e(k))) \tag{42}$$

Different from the non-time-delay case, the first two steps, $u(k-d)$ and $u(k-d+1)$ of the U_e^{OPT}, are applied to the Eq. (26). (shown in Fig. 4)

At the next time point $k+1$, the future error-tracking model is

$$\begin{aligned} e((k+1)+1) = \Phi e(k+1) + \Gamma u_e((k+1)-d) \\ + \bar{\Gamma} u_e((k+1)-d+1) \end{aligned} \tag{43}$$

Here, we note that $u_e(k-d+1)$ has already been obtained and the input in one interval is invariable. Therefore, predictive tracking-error vector will change to the form

$$\begin{aligned} E(k+1) = F(k+1)e(k+1) + G(k+1)U_e(k+1) \\ + \bar{G}(k+1)u_e(k-d+1) \end{aligned} \tag{44}$$

where

$$E(k+1) \triangleq \begin{bmatrix} e(k+2) \\ e(k+3) \\ \vdots \\ e(k+1+h) \end{bmatrix} \tag{45}$$

$$F(k) = \begin{bmatrix} \Phi(k+1) \\ \Phi(k+2)\Phi(k+1) \\ \vdots \\ \prod_{j=0}^{h-1} \Phi(k+1+j) \end{bmatrix},$$

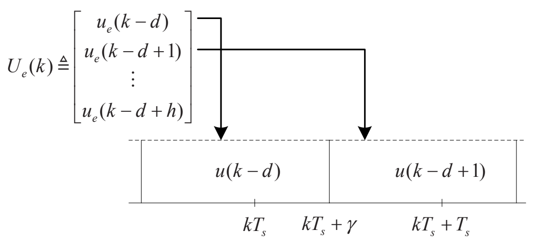

$U_e(k) \triangleq \begin{bmatrix} u_e(k-d) \\ u_e(k-d+1) \\ \vdots \\ u_e(k-d+h) \end{bmatrix}$
$u(k-d)$
$u(k-d+1)$
kT_s
$kT_s+\gamma$
kT_s+T_s

Fig. 4. Scheme of input with time-delay at current point.

$$U_e(k+1) \triangleq \begin{bmatrix} u_e(k-d+2) \\ u_e(k-d+3) \\ \vdots \\ u_e(k-d+h+1) \end{bmatrix} \tag{46}$$

Now, note that $U_e(k) \in \mathbb{R}^{m \cdot h}$, because $u_e(k-d+1)$ has been already known.
From (34), we get

$$\begin{aligned} \bar{G}(k+1) &= G_1(k+1) \\ G(k+1) &= \begin{bmatrix} G_2(k+1) & G_3(k+1) & \ldots & G_{h+1}(k+1) \end{bmatrix} \end{aligned} \tag{47}$$

The standard form of QP becomes

$$V(U_e(k+1)) = \bar{V} + \frac{1}{2} U_e(k+1)^T \cdot H \cdot U_e(k+1) + f \cdot U_e(k+1) \tag{48}$$

where

$$\begin{aligned} \bar{V} &= e(k+1)^T [F_r - F(k+1)]^T \bar{Q} [F_r - F(k+1)] e(k+1) \\ &\quad - 2[\bar{G}(k+1) u_e(k-d+1)]^T \bar{Q} [F_r - F(k+1)] e(k+1) \\ &\quad - [\bar{G}(k+1) u_e(k-d+1)]^T \bar{Q} [\bar{G}(k+1) u_e(k-d+1)] \\ H &= 2(G(k+1)^T \bar{Q} G(k+1) + \bar{R}) \\ f &= -2[G(k+1)^T \bar{Q} (F_r - F(k)) e(k+1) \\ &\quad + G(k+1)^T \bar{Q} \bar{G}(k+1) u_e(k-d+1)]^T \end{aligned} \tag{49}$$

The control sequence is

$$U_e^{OPT}(k+1) = arc\min(V(U_e(k+1))) \tag{50}$$

Then, the known input $u_e(k-d+1)$ and the first step $u_e(k-d+2)$ of $U_e^{OPT}(k+1)$ are applied to (44), as illustrated in Fig. 5.

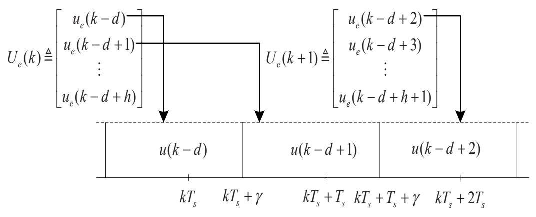


Fig. 5. Scheme of input with time-delay at next point.

5. Computer simulations and results

5.1 The complement of the constraints

During the control of a wheeled mobile robot, the bounded velocity and acceleration constraints are considered. The robot's tangential velocity υ and angular velocity ω are restricted as

$$\begin{aligned} &\upsilon < \upsilon_{\max}, \upsilon_{\max} = \tilde{\omega}_{\max} r \\ &\omega < \omega_{\max}, \omega_{\max} = 2\tilde{\omega}_{\max} r / L \end{aligned} \tag{51}$$

where $\tilde{\omega}_{\max}$ is the maximum angular velocity of each wheel, r is the radius of the wheel and L is the wheelbase of two driving wheels. A saturation of the command velocities that preserve the current curvature $\kappa = \dfrac{\omega}{\upsilon}$ is performed as (see [26])

$$\sigma = \max(|\upsilon| / \upsilon_{\max}, |\omega| / \omega_{\max}, 1) \tag{52}$$

where the actual command velocities υ_c and ω_c stand for

$$\begin{aligned} &\upsilon_c = sign(\upsilon)\upsilon_{\max}, \omega_c = \omega/\sigma, && \text{if } \sigma = |\upsilon| / \upsilon_{\max} \\ &\upsilon_c = \upsilon/\sigma, \omega_c = sign(\omega)\omega_{\max}, && \text{if } \sigma = |\omega| / \omega_{\max} \\ &\upsilon_c = \upsilon, \omega_c = \omega, && \text{if } \sigma = 1 \end{aligned} \tag{53}$$

In the simulations, the maximum allowed tangential velocity and angular velocity were set as follows:

$$\upsilon_{\max} = 0.5m/s, \omega_{\max} = 5rad/s \tag{54}$$

5.2 The implementation of error-tracking control

The parameters are selected as follows:

$m = 5kg$, $I = 0.05kg \cdot m^2$, $L = 0.2m$, $r = 0.03m$, $T_s = 0.1s$, $h = 4$, $D = 0.56\ s$, $\gamma = 0.06\ s$

$$A_r = \begin{bmatrix} 0.65 & 0 & 0 \\ 0 & 0.65 & 0 \\ 0 & 0 & 0.65 \end{bmatrix}, \quad R = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix},$$

$$Q = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 40 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}.$$

The ratio of the diagonal elements in Q determines the sensitivity of the resulting controller to a certain error. A higher value of the diagonal element increases the sensitivity to the corresponding error. In the present case the control for the error in the lateral direction of driving has the highest weight, while the control for the orientation error has the highest sensitivity. Similarly, the diagonal elements in R define the energy of the input-velocity signals; the lowest value of the elements results in more energy-consuming control.

5.3 The eight-shaped trajectory tracking

The eight-shaped trajectory is defined as follows:

$$x_r(t) = 0.8\sin(\frac{4\pi}{30}t)$$
$$y_r(t) = \sin(\frac{2\pi}{30}t)$$

Firstly, we considered a nonlinear optimal control method. Applying the Euler's approximation to (2), we obtain the following discrete-time model for the robot motion:

$$q(k+1) = q(k) + T_s \begin{bmatrix} \cos\theta(k) & 0 \\ \sin\theta(k) & 0 \\ 0 & 1 \end{bmatrix} u(k),$$
$$q(k+1) = \begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix}, q(k) = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix}, u(k) = \begin{bmatrix} \upsilon(k) \\ \omega(k) \end{bmatrix}$$

The tracking control policy is derived from minimizing a cost function, which penalizes the predictive position error and control signals:

$$J = \frac{1}{2}\left\{[q_r(k+1) - q(k+1)]^T Q[q_r(k+1) - q(k+1)] + u^T(k)Ru(k)\right\}$$

The start pose is chosen as $[-0.4, 0.1, 0]^T$. The mobile robot trajectory tracking results, obtained by such nonlinear optimal approach, are shown in Fig. 6. Meanwhile, the tracking performance by the time-free RHC is obtained in Fig. 7(a). Although the mobile robot's convergence is fast to the reference trajectory in both cases, it is obvious that the RHC performs better than the former method.

Then, the time-delay tracking performance of the robot with RH controller is shown in Fig. 8(a). We can see that the time-delay case is almost as good as the time-free one. The delay control signals are illustrated in Fig. 8(b) corresponding to Fig. 7(b). Compared to the error states in Fig. 7(c), the time-delay case Fig. 8(c) is a little worse, where the fluctuations are mostly present in the orientation datum. However, the errors converge to zero approximately quickly and the tiny fluctuations are in the accepted bound. Finally, the total torques acting on two driving wheels are shown in Fig. 8(d).

The second trajectory in the test is a circle defined as follows:

$$x_r = 1.5\cos(\frac{2\pi t}{30})$$
$$y_r = 1.5\sin(\frac{2\pi t}{30})$$

The initial pose is selected for the tests: $[1.6, -0.5, 1/6\pi]^T$. The time-free and time-delay results by RHC are shown as follows:

Fig. 10(a) indicates that delay tracking performance is just worse than Fig. 9(a) during the beginning response. Fig. 10(b) shows that within the delay time district at the beginning, the tangential velocity and angular velocity are zero. And after the dead time, the input is applied to the tracking system. The error states are close to zero perfectly after $5\,s$ in Fig. 10(c) and the driving torques are obtained in Fig. 10(d).

From the simulation results, we can see that the RH controller gives good control results, which is to be expected because of the more complex control structure, and taking into account future values of the reference within a certain predictive horizon period.

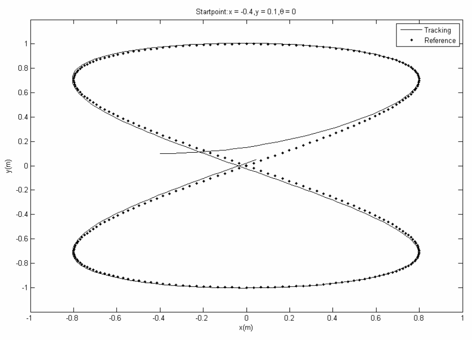
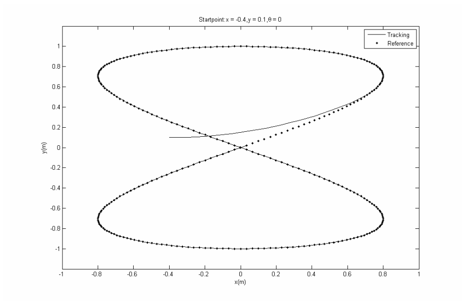
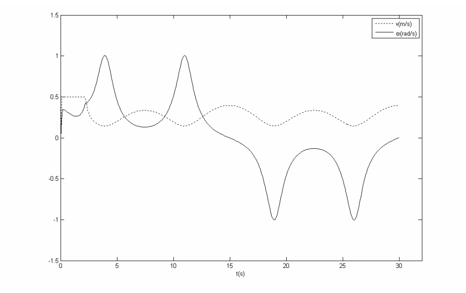


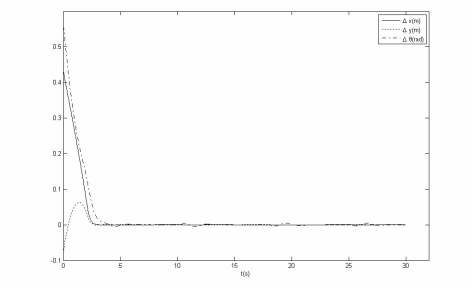
Fig. 6. Time-free trajectory tracking with the nonlinear optimal controller: robot path and reference path.



(a) Time-free trajectory tracking with the RH controller: robot path reference path.

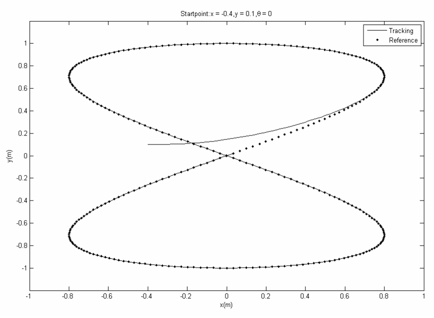


(b) Time-free trajectory tracking with the RH controller: and tangential velocity and angular velocity.

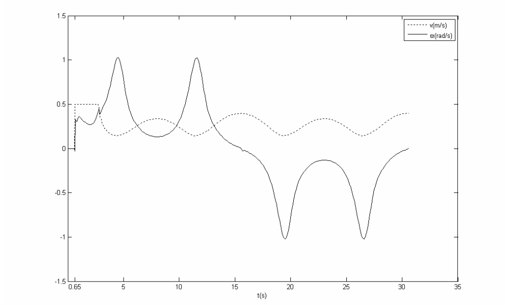


(c) Time-free trajectory tracking with the RH controller: errors of three system states.

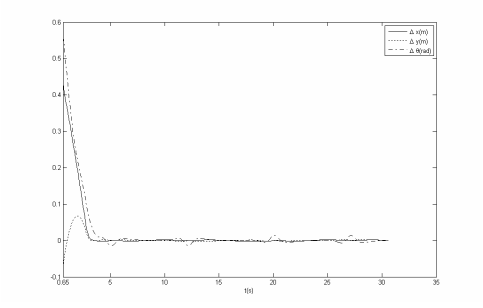
Fig. 7.



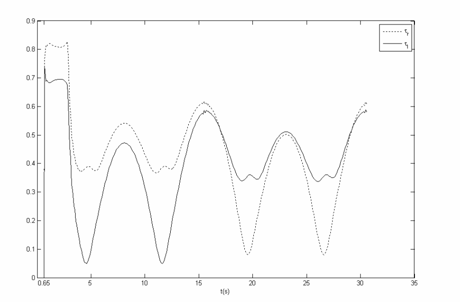
(a) Time-delay trajectory tracking with the RH controller: robot path and reference path.



(b) Time-delay trajectory tracking with the RH controller: tangential velocity and angular velocity

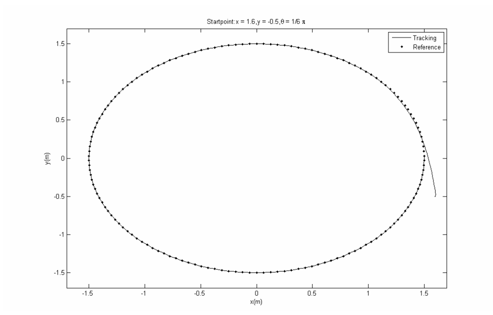


(c) Time-delay trajectory tracking with the RH controller: errors of three system states.

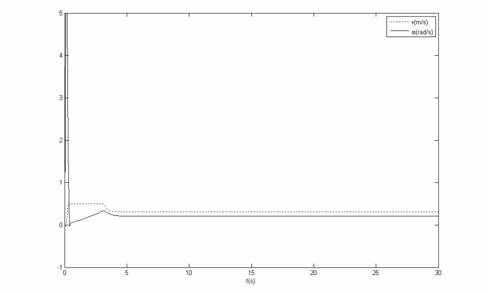


(d) Time-delay trajectory tracking with the RH controller: torques on right and left driving wheels.

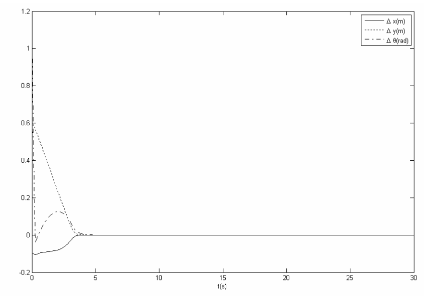
Fig. 8.



(a) Time-free trajectory tracking with the RH controller: robot path and reference path.

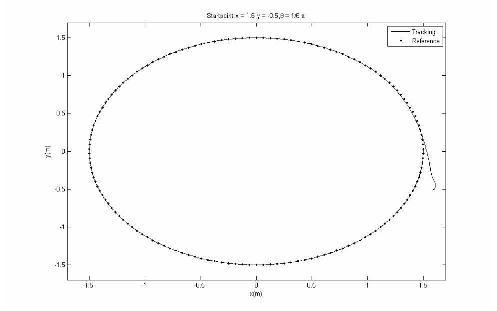


(b) Time-free trajectory tracking with the RH controller: tangential velocity and angular velocity.

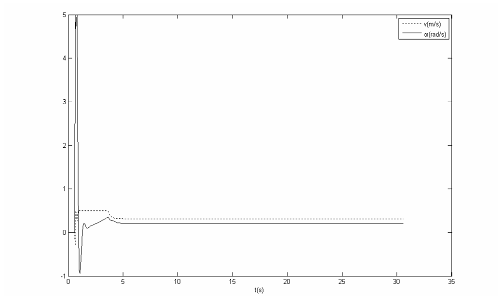


(c) Time-free trajectory tracking with the RH controller: errors of three system states.

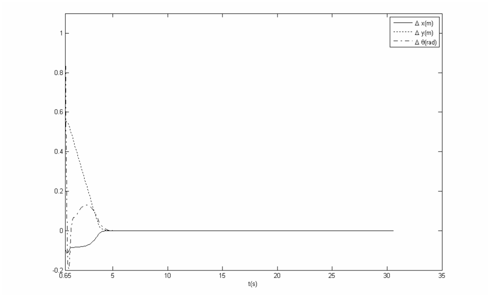
Fig. 9.



(a) Time-delay trajectory tracking with the RH controller: robot path and reference path.



(b) Time-delay trajectory tracking with the RH controller: tangential velocity and angular velocity.



(c) Time-delay trajectory tracking with the RH controller: errors of three system states.

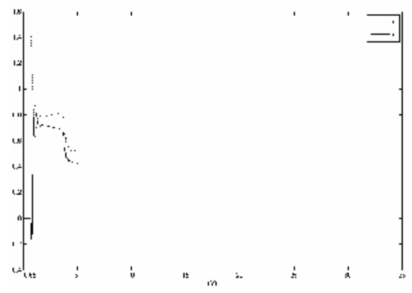

(d) Time-delay trajectory tracking with the RH controller: torques on right and left wheels

Fig. 10.

6. Conclusions

The receding horizon (RH) controller for time-delay error-tracking control of differentially steered wheeled mobile robots is presented in this paper. The control policy is derived from the optimization of the quadratic cost function, which penalizes the tracking error and control variables in each sampling time. And the minimizing problem is solved by using the QP method, taking the current error state as the initial value and including the constraints. Here, the tracking-error kinematics model is first linearized at the equilibrium point. And then, it is transferred to an exact discrete model with delay. The proposed controller also includes velocity and acceleration constraints to prevent the mobile robot from slipping.

The simulation results demonstrate the performance of the receding horizon error-tracking control of wheeled mobile robots. Compared to the time-free cases, they show that even with the time-delay, the robot converges to the trajectory accurately and fast. The RHC method is confirmed to be a very feasible one to control the wheeled robots.

References

[1] F. Pourboghrat, Exponential stabilization of nonholonomic mobile robots, *Computers and Electrical Engineering*. 28 (5) (2002) 349-359.

[2] D. Gu and H. Hu, A stabilizing receding horizon regulator for nonholonomic mobile robots, *IEEE Transactions on Robotics*, 21 (5) (2005) 1022-1028.

[3] K. C. Park, H. Chung and J. G. Lee, Point stabilization for mobile robots via state-space exact feedback linearization, *Robotics and Computer Integrated Manufacturing*. 16 (5) (2000) 353-363.

[4] K. T. Song and C. E. Li, Tracking control of a fee ranging automatic guided vehicle, *Control Engineering Practice*, 1 (1993) 163-169.

[5] Z. P. Jiang and H. Nijmeijer, Tracking control of mobile robots: a case study in backstepping, *Automatica*, 33 (7) (1997) 1393-1399.

[6] M. Fliess, J. Levine, P. Martin and P. Rouchon, Design of trajectory stabilizing feedback for driftless flat systems, *Proceedings of European Control Conference*. (1997) 1882-1887.

[7] R. T. M'Closkey and R. M. Murray, Exponential stabilization od nonlinear driftless control systems via time-varying homogeneous feedback, Proc. *IEEE Conf. Decision Contr*. (1994) 1317-1322.

[8] Z. P. Jiang, E. Lefeber and H. Nijmeijer, Saturated stabilization and tracking of a nonholonomic mobile robot, *Systems and control letters*. 42 (5) (2001) 327-332.

[9] T. C. Lee, K. T. Song, C. H. Lee and C. C. Teng, Tacking control of unicycle-model mobile robots using a saturation feedback controller, *IEEE Transaction on Control Systems technology*. 9 (2001) 1305-318.

[10] D. H. Kim and J. H. Oh, Tracking control of two-wheeled mobile robot using input-output linearization, *Control Engineering Practice*. 7 (3) (1999) 369-373.

[11] K. D. Do, J. Pan, Global output-feedback path tracking of unicycle-type mobile robots, *Robotics and Computer-Integrated Manufacturing*, 22 (2) (2006) 166-179.

[12] T. L. Chung, T. T. Nguyen and S. B. Kim, Sliding Mode Control of Two-Wheeled Welding Mobile Robot for Tracking Smooth Curved Welding Path, *KSME International Journal*. 18 (7) (2004) 1094-1106.

[13] T. H. Bui, T. L. Chung, S. B. Kim and T. T. Nguyen, Adaptive Tracking Control of Two-Wheeled Welding Mobile Robot with Smooth Curved Welding Path, *KSME International Journal*. 17 (11) (2003) 1682-1692.

[14] T. L. Chung, T. H. Bui, S. B. Kim, M. S. Oh and T. T. Nguyen, Wall-Following Control of a Two-Wheeled Mobile Robot, *KSME International Journal*. 18 (8) (2004) 1288-1296.

[15] M. G. Na, D. W. Jung, S. H. Shin, S. M. Lee, Y. J. Lee, J. W. Jang and K. B. Lee, Design of a Nuclear Reator Controller Using a Model Predictive Control Method, *KSME International Journal*. 18 (12) (2004) 2080-2094.

[16] N. Sheibat-Othman, O. Garrigues and S. Othman, Receding horizon control for polymerization processes, *Mathematical and Computer Modelling*. 46 (1-2) (2007) 251-259.

[17] D. Nešić and L. Grüne, A receding horizon control approach to sampled-data implementation of continuous-time controllers, *Systems & Control Letters*. 55 (8) (2006) 660-672.

[18] G. Klančar and I. Škrjanc, Tracking-error model-based predictive control for mobile robots in real time, *Robotics and Autonomous Systems*. 55 (6) (2007) 460-469.

[19] K. B. Kim and J. W. Choi, Game Optimal Receding Horizon Guidance Laws and Its Equivalence to Receding Horizon Guidance Laws, *KSME Interna-*

tional Journal. 16 (6) (2002) 770-775.

[20] J. E. Normey-Rico, J. Gomez-Ortega, E. F. Camacho, A Smith-predictor-based generalised predictive controller for mobile robot path-tracking, *Control Engineering Practice*. 7 (6) (1999) 729–740.

[21] D. Gu and H. Hu, Neural predictive control for a car-like mobile robot, *International Journal of Robotics and Autonomous Systems*. 39 (2) (2002) 2-3.

[22] Y. J. Kanayama, Y. Kimura, F. Miyazaki and T. Noguchi, A stable tracking control method for an autonomous mobile robot, *in Proc. IEEE Int. Conf. Robot. Autom*. (1990) 384–389.

[23] F. L. Lewis, C. T. Abdallah and D. M. Dawson, *Control of Robot Manipulators*, New York: MacMillan, 1993.

[24] F. E. Udwadia and R. E. Kalaba, A new perspective on constrained motion, *Mathematical and Physical Sciences*. 439 (1906) (1992) 407-410.

[25] G. C. Goodwin, M. M. Seron and J. A. De Doná, *Constraint Control and Estimation*, Springer-Verlag London Limited 2005.

[26] G. Oriolo, A. De Luca and M. Vendittelli, WMR control via dynamic feedback linearization: design, implementation, and experimental validation, *IEEE Transactions on Control Systems Technology*. 10 (6) (2002) 835-852.

Appendix

Appendix. A:

The time derivative of Eq. (12) is calculated as follows:

$$\begin{aligned}\dot{x}_e &= (\dot{x}_r-\dot{x})\cos\theta+(\dot{y}_r-\dot{y})\sin\theta\\ &\quad -(x_r-x)\dot{\theta}\sin\theta+(y_r-y)\dot{\theta}\cos\theta\\ &= y_e\omega-\upsilon+\dot{x}_r\cos\theta+\dot{y}_r\sin\theta\\ &= y_e\omega-\upsilon+\dot{x}_r\cos(\theta_r-\theta_e)+\dot{y}_r\sin(\theta_r-\theta_e)\\ &= y_e\omega-\upsilon+\dot{x}_r(\cos\theta_r\cos\theta_e+\sin\theta_r\sin\theta_e)\\ &\quad+\dot{y}_r(\sin\theta_r\cos\theta_e-\cos\theta_r\sin\theta_e)\\ &= y_e\omega-\upsilon+(\dot{x}_r\cos\theta_r+\dot{y}_r\sin\theta_r)\cos\theta_e\\ &\quad+(\dot{x}_r\sin\theta_r-\dot{y}_r\cos\theta_r)\sin\theta_e\\ &= \omega y_e-\upsilon+\upsilon_r\cos\theta_e\\ \dot{y}_e &= -(\dot{x}_r-\dot{x})\sin\theta+(\dot{y}_r-\dot{y})\cos\theta\\ &\quad -(x_r-x)\dot{\theta}\cos\theta+(y_r-y)\dot{\theta}\sin\theta\\ &= -x_e\omega+\dot{x}\sin\theta-\dot{y}\cos\theta-\dot{x}_r\sin\theta+\dot{y}_r\cos\theta\\ &= -x_e\omega-\dot{x}_r\sin(\theta_r-\theta_e)+\dot{y}_r\cos(\theta_r-\theta_e)\\ &= -x_e\omega-\dot{x}_r(\sin\theta_r\cos\theta_e-\cos\theta_r\sin\theta_e)\\ &\quad+\dot{y}_r(\cos\theta_r\cos\theta_e+\sin\theta_r\sin\theta_e)\\ &= -x_e\omega+(\dot{x}_r\cos\theta_r+\dot{y}_r\sin\theta_r)\sin\theta_e\\ &\quad+(\dot{y}_r\cos\theta_r-\dot{x}_r\sin\theta_r)\cos\theta_e\\ &= -x_e\omega+\upsilon_r\sin\theta_e\\ \dot{\theta}_e &= \dot{\theta}_r-\dot{\theta}=\omega_r-\omega\end{aligned}$$

Appendix. B:

From Eq. (15), if

$$\dot{e}=f(\tilde{e},\tilde{u}_e)=0$$

when

$$\tilde{e}=[0,0,0]^T,\tilde{u}_e=[0,0]^T$$

$(\tilde{e},\tilde{u}_e)$ is the equilibrium point of Eq. (15).

Using Taylor's series, we obtain

$$f(e,u_e)=f(\tilde{e},\tilde{u}_e)+\left.\frac{\partial f}{\partial e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}(e-\tilde{e})+\left.\frac{\partial f}{\partial u_e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}(u-\tilde{u}_e)$$
$$+\text{high-order-terms}$$
$$f(e,u_e)\approx f(\tilde{e},\tilde{u}_e)+\left.\frac{\partial f}{\partial e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}(e-\tilde{e})+\left.\frac{\partial f}{\partial u_e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}(u-\tilde{u}_e)$$
$$\approx\left.\frac{\partial f}{\partial e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}e+\left.\frac{\partial f}{\partial u_e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}u_e$$

$$\left.\frac{\partial f}{\partial e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}=\begin{bmatrix}0&\omega_r&0\\-\omega_r&0&\upsilon_r\\0&0&0\end{bmatrix},\quad \left.\frac{\partial f}{\partial u_e}\right|_{\substack{e=\tilde{e}\\u_e=\tilde{u}_e}}=\begin{bmatrix}-1&0\\0&0\\0&-1\end{bmatrix}$$

Kil To Chong (M'96) received the Ph.D. degree in mechanical engineering from Texas A&M University, College Station, in 1995. Currently, he is a Professor at the School of Electronics and Information Engineering, Chonbuk National University, Jeonju, Korea, and Head of the Mechatronics Research Center granted from the Korea Science Foundation. His research interests are in the areas of motor fault detection, network system control, time-delay systems, and neural networks.

Chang Goo Lee was born in Chonju, South Korea on Dec., 1958. He received the B.S. and M.S., and Dr.Eng. degrees in Electrical Engineering from Chonbuk National University, South Korea, 1981, 1983 and 1990 respectively. He had been with ETRI as a senior researcher from 1983 to 1991. Since 1992, He has been with the School of Electronic and Information Engineering, Chonbuk National University where he is presently a Professor. His research interests include intelligent control, nonlinear control, and home network control.

Yu Gao received the master's degree in Electronics and Information from Chonbuk National University, Korea, in 2008. He got his bachelor's degree in Physics from Soochow University, China, in 2005. Currently, he is a Ph.D. candidate in the School of Electronics and Information, Chonbuk National University, Korea. His research interests are in the area of the receding horizon control.