A Stable Tracking Control Method for an Autonomous Mobile Robot

Yutaka Kanayama

Dept of Computer Science
Naval Postgraduate School
Monterey, CA 93943

CORI/Dept of Computer Science
University of California
Santa Barbara, CA 93106

Yoshihiko Kimura

Center for Robotic Systems
University of California
Santa Barbara, CA 93106

Mitsubishi Metal Corp.
Tokyo, Japan

Fumio Miyazaki

Osaka University
Dept of Mechanical Engineering
Toyonaka, Japan

Tetsuo Noguchi

Center for robotic Systems
University of California
Santa Barbara, CA 93106

ABSTRACT

This paper proposes a stable tracking control rule for non-holonomic vehicles. Stability of the rule is proved through the use of a Liapunov function. Input to the vehicle are a reference posture $(x_r, y_r, \theta_r)^t$ and reference velocities $(v_r, \omega_r)^t$. The major objective of this paper is to propose a control rule to find a reasonable target linear and rotational velocities $(v, \omega)^t$. Linearizing the system's differential equation is useful to decide parameters for critical dumping for a small disturbance. In order to avoid any slippage, a velocity/acceleration limitation scheme was introduced. Several simulation results are presented with or without the velocity/acceleration limiter. The control rule and limiting method proposed in this paper are robot-independent and hence can be applied to various kind of mobile robots with a dead reckoning ability. This method was implemented on the autonomous mobile robot Yamabico-11. Experimental results obtained are close to the results with the velocity/acceleration limiter.

1. Introduction

The purpose of this paper is to propose a stable tracking control method for a non-holonomic vehicle with abundant simulation results. Real experimental results on the autonomous mobile robot Yamabico-11 are also presented.

Tsumura proposed a method in which the reference point sequence is stored in memory. In each cycle of the locomotion control, the reference point and the future position of the robot is compared for determining the next steering [2]. Kanayama proposed a method using straight line reference for the robot's locomotion instead of a sequence of points [3]. Its velocity and steering control method has some similarities to the one proposed in this paper. Crowley developed a locomotion control system whose organization has a three layered structure [4]. He defines the concept of "virtual vehicle" which is useful for constructing a system which is robot independent. In its command system, independent control of linear and rotational motion is possible, thus enabling smooth clothoid curves [5]. Singh used an inverse kinematic and a quintic polynomial method for compensating errors in vehicle tracking [6]. In the second method, he interpolates the current point and a future reference point with a smooth curve.

Kanayama proposed the use of a reference and current postures for vehicle control, the use of a local error coordinate system, and a PI control algorithm for linear/rotational velocity rules in an earlier locomotion control method on the Yamabico-11 robot [7]. Nelson proposed a locomotion control method for a cart with a front steering wheel, in which they also used the error coordinate system [8]. They adopted a linear function in control rules for steering and linear velocity. These two papers are regarded as pioneers of this paper.

In this paper, a new control rule for determining vehicle's linear and rotational velocities are given, which are different from both of [7] and [8]. The stability of the control rule is proven using a Liapunov function [9][10][11]. The use of the trace function $(1 - \cos\theta)$ of orientation θ is successful in finding an appropriate Liapunov function [11]. One of the difficulties of this problem lies in the fact that ordinary vehicles possess only two degrees of freedom (linear velocity v and rotational velocity ω) for locomotion control, although vehicles have three degrees of freedom, x, y and θ in its positioning. Another difficulty is in the non-linearity of the kinematic relation between $(v, \omega)^t$ and $(\dot{x}, \dot{y}, \dot{\theta})^t$. The use of a Liapunov function resolves these difficulties.

By linearizing the system's differential equation, we find a condition for critical dumping, which gives appropriate parameters for specific control rules. The need of velocity/acceleration limitation is also discussed. After these analyses and discussions, abundant simulation results are presented. The method described so far is hardware independent and applicable to ordinary (not omni-directional) vehicles.

This method is useful to the class of autonomous vehicles in which (a) a dead reckoning capability is provided, (b) reference path specification and current position estimation (through dead reckoning) are given separately, and (c) high precision in positional control is mandatory. This method was implemented on the autonomous mobile robot Yamabico-11 which has been developed at the University of Tsukuba, the University of California at Santa Barbara, and Naval Postgraduate School. It was demonstrated that these algorithms are sound and provided precise tracking control. An extensive set of the experimental results are shown.

2. Problem Statements

Before stating the problem, we will give a few preliminary definitions.

2.1. Path Representation and Vehicle Kinematics

There is a mobile robot which is located on a 2D plane in which a global Cartesian coordinate system is defined. The robot in the world

CH2876-1/90/0000/0384$01.00 © 1990 IEEE

possesses three degrees of freedom in its positioning which are represented by a *posture*,

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \tag{1}$$

where the heading direction θ is taken counterclockwise from the x-axis. Let $\mathbf{0}$ denote a *null posture* $(0, 0, 2n\pi)^t$, where n is an integer. Since the robot has a locomotion capability in the plane, the posture $\mathbf{p}$ is in fact a function of time t. The entire locus of the point $(x(t), y(t))$ is called a *path* or *trajectory*. If the time derivatives $\dot{x}$ and $\dot{y}$ exist, $\theta(t)$ is not an independent variable any more, because

$$\theta(t) = \tan^{-1}\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \tag{2}$$

The vehicle's motion is controlled by its *linear velocity* v and *rotational velocity* ω, which are also functions of time. The vehicle's kinematics is defined by a Jacobian matrix J:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \dot{\mathbf{p}} = J\mathbf{q} \equiv \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \mathbf{q} \tag{3}$$

where $\mathbf{q} = (v, \omega)^t$. This kinematics is common to all kinds of vehicles which are not omni-directional. (For instance, an automobile, a bicycle, a vehicle with two parallel independent power wheels - power wheeled steering system, and a tricycle) The linear velocity v and rotational velocity ω of this kind of vehicle is controlled by its accelerator and steering wheel or handle respectively.

2.2. Error Posture

In this control system, two postures are used; the *reference posture* $\mathbf{p}_r = (x_r, y_r, \theta_r)^t$ and the *current posture* $\mathbf{p}_c = (x_c, y_c, \theta_c)^t$. A reference posture is a goal posture of the vehicle and a current posture is its "real" posture at this moment respectively (Fig. 1). We will define an *error posture* $\mathbf{p}_e$ of the two, which is a transformation of the reference posture $\mathbf{p}_r$ in a local coordinate system with an origin of (x_c, y_c) and an X-axis in the direction of θ_c [7][8] (Fig. 2). This is the "difference" between $\mathbf{p}_r$ and $\mathbf{p}_c$:

$$\mathbf{p}_e = \begin{pmatrix} x_e \\ y_e \\ \theta_e \end{pmatrix} = \begin{pmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{p}_r - \mathbf{p}_c) \equiv T_e(\mathbf{p}_r - \mathbf{p}_c) \tag{4}$$

If $\mathbf{p}_r = \mathbf{p}_c$, the error posture $\mathbf{p}_e = \mathbf{0}$. If $\mathbf{p}_r$ is ahead of $\mathbf{p}_c$ (the vehicle is behind of the goal), $x_e > 0$. For instance, if $\mathbf{p}_c = (3/2, 1, \pi/6)$ and $\mathbf{p}_r = (5/2, 1+\sqrt{3}, \pi/4)$, $\mathbf{p}_e = (\sqrt{3}, 1, \pi/12)$ (Figure 2 illustrates this case).

2.3. Problem

Now, we are able to state the architecture of a tracking control system for the vehicle (Fig. 3). The global input of the system is the reference posture $\mathbf{p}_r$ and *reference velocities* $\mathbf{q}_r = (v_r, \omega_r)^t$, which are variables of time. The global output of the system is the current posture $\mathbf{p}_c$. The purpose of this tracking controller is to converge the error posture to $\mathbf{0}$. Let us describe each component in Figure 3 from left to right. The first component calculate an error posture from $\mathbf{p}_r$ and $\mathbf{p}_c$ using Equation (4). The second box is a control rule for the vehicle, which calculates a *target velocities* $\mathbf{q} = (v, \omega)^t$ using the error posture $\mathbf{p}_e$ and the reference velocities $\mathbf{q}_r = (v_r, \omega_r)$:

$$\mathbf{q} = \begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} v(\mathbf{p}_e, \mathbf{q}_r) \\ \omega(\mathbf{p}_e, \mathbf{q}_r) \end{pmatrix} \tag{5}$$

The third box T stands for the vehicle hardware capability of transforming target velocities to vehicle's real current velocities. In Sections 3 and 4,

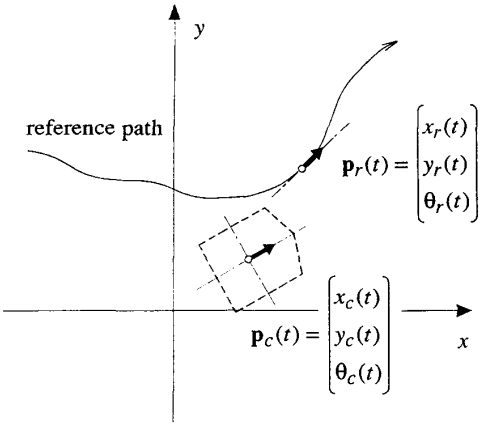

Fig. 1 Reference and Current Postures

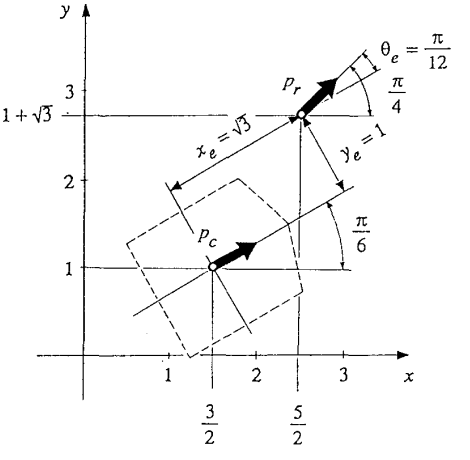

Fig. 2 Error Posture

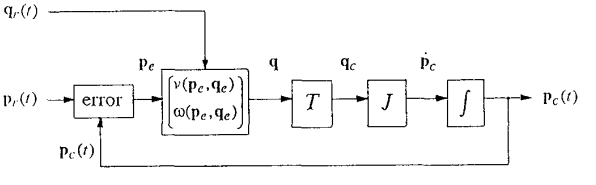

Fig. 3 Architecture of Tracking Controller

specifically, we assume the identity transformation:

$$\mathbf{q}_c = \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{q} \tag{6}$$

This *perfect velocity tracking assumption* simplifies the forthcoming analysis.

The fourth box is the kinematics matrix M in Equation (3) to produce the derivative of a current posture $\mathbf{p}_c$. The last box is for integration. Thus, only unknown component in this system is the control rule. Since the system's input $\mathbf{p}_r$ is time-variable, it is called "non-autonomous" by the definition in the control theory [9].

3. A Control Scheme and Its Stability

In this section, we will find a stable control rule using a Liapunov function [9]. The following lemma follows the system depicted in Figure 1.

Lemma 1.

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \dot{\mathbf{p}}_e = \mathbf{f}(t,\ \mathbf{p}_e) = \begin{bmatrix} \omega(\mathbf{p}_e,\ \mathbf{q}_r)\ y_e - v\ (\mathbf{p}_e,\ \mathbf{q}_r) + v_r \cos\theta_e \\ -\omega(\mathbf{p}_e,\ \mathbf{q}_r)\ x_e + v_r \sin\theta_e \\ \omega_r - \omega(\mathbf{p}_e,\ \mathbf{q}_r) \end{bmatrix} \tag{7}$$

Proof. Using Equation (4) and an equality $\dot{x}_r \sin\theta_r = \dot{y}_r \cos\theta_r$ from Equation (3),

$$\begin{aligned}
\dot{x}_e &= (\dot{x}_r - \dot{x}_c)\cos\theta_c + (\dot{y}_r - \dot{y}_c)\sin\theta_c - (x_r - x_c)\dot{\theta}_c \sin\theta_c + (y_r - y_c)\dot{\theta}_c \cos\theta_c \\
&= y_e\omega_c - v_c + \dot{x}_r\cos\theta_c + \dot{y}_r\sin\theta_c \\
&= y_e\omega_c - v_c + \dot{x}_r\cos(\theta_r - \theta_e) + \dot{y}_r\sin(\theta_r - \theta_e) \\
&= y_e\omega_c - v_c + \dot{x}_r(\cos\theta_r\cos\theta_e + \sin\theta_r\sin\theta_e) + \dot{y}_r(\sin\theta_r\cos\theta_e - \cos\theta_r\sin\theta_e) \\
&= y_e\omega_c - v_c + (\dot{x}_r\cos\theta_r + \dot{y}_r\sin\theta_r)\cos\theta_e + (\dot{x}_r\sin\theta_r - \dot{y}_r\cos\theta_r)\sin\theta_e \\
&= y_e\omega_c - v_c + v_r\cos\theta_e
\end{aligned}$$

$$\begin{aligned}
\dot{y}_e &= -(\dot{x}_r - \dot{x}_c)\sin\theta_c + (\dot{y}_r - \dot{y}_c)\cos\theta_c - (x_r - x_c)\dot{\theta}_c \cos\theta_c - (y_r - y_c)\dot{\theta}_c \sin\theta_c \\
&= -x_e\omega_c + \dot{x}_c\sin\theta_c - \dot{y}_c\cos\theta_c - \dot{x}_r\sin\theta_c + \dot{y}_r\cos\theta_c \\
&= -x_e\omega_c - \dot{x}_r\sin(\theta_r - \theta_e) + \dot{y}_r\cos(\theta_r - \theta_e) \\
&= -x_e\omega_c - \dot{x}_r(\sin\theta_r\cos\theta_e - \cos\theta_r\sin\theta_e) + \dot{y}_r(\cos\theta_r\cos\theta_e + \sin\theta_r\sin\theta_e) \\
&= -x_e\omega_c + (\dot{x}_r\cos\theta_r + \dot{y}_r\sin\theta_r)\sin\theta_e + (\dot{y}_r\cos\theta_r - \dot{x}_r\sin\theta_r)\cos\theta_e \\
&= -x_e\omega_c + v_r\sin\theta_e
\end{aligned}$$

$$\dot{\theta}_e = \dot{\theta}_r - \dot{\theta}_c = \omega_r - \omega_c$$

Substituting v_c and ω_c by $v(\mathbf{p}_e,\ \mathbf{q}_r)$ and $v(\mathbf{p}_e,\ \mathbf{q}_r)$ respectively (cf. Equations (6) and (5)), we obtain the lemma. □

Let us propose a specific instance of the control rule (5) for the target velocities as follows:

$$\mathbf{q} = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v(\mathbf{p}_e,\ \mathbf{q}_r) \\ \omega(\mathbf{p}_e,\ \mathbf{q}_r) \end{bmatrix} = \begin{bmatrix} v_r\cos\theta_e + K_x x_e \\ \omega_r + v_r(K_y y_e + K_\theta \sin\theta_e) \end{bmatrix} \tag{8}$$

where K_x, K_y and K_θ are positive constants. The first term in each velocity is a feedforward part. By Lemma 1:

Lemma 2.

$$\dot{\mathbf{p}}_e = \mathbf{f}(t,\ \mathbf{p}_e) = \begin{bmatrix} (\omega_r + v_r(K_y y_e + K_\theta \sin\theta_e))\ y_e - K_x x_e \\ -(\omega_r + v_r(K_y y_e + K_\theta \sin\theta_e))\ x_e + v_r\sin\theta_e \\ -v_r(K_y y_e + K_\theta \sin\theta_e) \end{bmatrix} \tag{9}$$

Soundness of this control rule (8) is established by the following proposition:

Proposition 1. If we use the control rule (8), $\mathbf{p}_e = \mathbf{0}$ is a stable equilibrium point if the reference velocity $v_r > 0$.

Proof. Let us propose a scalar function V as a Liapunov function candidate [9]:

$$V = \frac{1}{2}(x_e^2 + y_e^2) + (1 - \cos\theta_e)/K_y \tag{10}$$

Clearly, $V \geq 0$. If $\mathbf{p}_e = \mathbf{0}$, $V = 0$. If $\mathbf{p}_e \neq \mathbf{0}$, $V > 0$. Furthermore, by Lemma 2:

$$\begin{aligned}
\dot{V} &= \dot{x}_e x_e + \dot{y}_e y_e + \dot{\theta}_e \sin\theta_e / K_y \\
&= [(\omega_r + v_r(K_y y_e + K_\theta \sin\theta_e))y_e - K_x x_e]\, x_e \\
&\quad + [-(\omega_r + v_r(K_y y_e + K_\theta \sin\theta_e))x_e + v_r\sin\theta_e]\, y_e \\
&\quad + [-v_r(K_y y_e + K_\theta \sin\theta_e)]\, \sin\theta_e / K_y \\
&= -K_x x_e{}^2 - v_r K_\theta \sin^2\theta_e / K_y \leq 0
\end{aligned}$$

Then, V becomes a Liapunov function. □

The following proposition demonstrates that the uniformly asymptotically stability around $\mathbf{p}_e = \mathbf{0}$ under some conditions:

Proposition 2. Assume that (a) v_r and ω_r are continuous, (b) v_r, ω_r, K_x, and K_θ are bounded, and (c) $\dot{v}_r$ and $\dot{\omega}_r$ are sufficiently small. Under these conditions, $\mathbf{p}_e = \mathbf{0}$ is uniformly asymptotically stable over $[0, \infty)$.

Proof. By linearizing the differential Equation (9) around $\mathbf{p}_e = \mathbf{0}$:

$$\dot{\mathbf{p}}_e = A\ \mathbf{p}_e \tag{11}$$

where

$$A = \begin{bmatrix} -K_x & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & -v_r K_y & -v_r K_\theta \end{bmatrix} \tag{12}$$

Then, $A(.)$ is continuously differentiable and is bounded. The characteristic equation for A is:

$$a_3 s^3 + a_2 s^2 + a_1 s + a_0 = 0 \tag{13}$$

where

$$\begin{cases} a_3 = 1 \\ a_2 = K_\theta v_r + K_x \\ a_1 = K_y v_r^2 + K_x K_\theta v_r + \omega_r^2 \\ a_0 = K_x K_y v_r^2 + \omega_r^2 K_\theta v_r \end{cases} \tag{13'}$$

Since all coefficients a_i are positive and $a_1 a_2 - a_0 a_3 > 0$, the real parts of all roots are negative through the Routh-Hurwitz Criterion. Therefore, by Corollary 41 on page 223 in [9], the Proposition was proved. □

4. Effects of Control Parameters

In the previous section, we demonstrated that the system is stable for any combination of parameter values of K_x, K_y, and K_θ. However, since we need a non-oscillatory, but not too slow response of the robot, we have to find an optimal parameter set. In order to simplify the analysis, we consider only situations in which the reference posture is moving on the x axis to the positive direction at a constant velocity V_r:

$$\mathbf{p}_r(t) = \begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} V_r t \\ 0 \\ 0 \end{bmatrix} \text{ and } \mathbf{q}_r(t) = \begin{bmatrix} v_r(t) \\ \omega_r(t) \end{bmatrix} = \begin{bmatrix} V_r \\ 0 \end{bmatrix} \tag{14}$$

This condition is called the *linear reference motion.* In addition, we assume that:

$$|\theta_e| \ll 1 \text{ and } |\dot{\theta}_e| \ll 1 \tag{15}$$

Proposition 3. Under Conditions (14) and (15),

$$\dot{\mathbf{p}}_c = \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} -K_x & 0 & 0 \\ 0 & 0 & V_r \\ 0 & -V_r K_y & -V_r K_\theta \end{bmatrix} \begin{bmatrix} x_c - V_r t \\ y_c \\ \theta_c \end{bmatrix} + \begin{bmatrix} V_r \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

Proof. By substituting Equation (11) by Equation (4),

$$\dot{\mathbf{p}}_c = T_e^{-1} (A\, T_e - \dot{T}_e)(\mathbf{p}_c - \mathbf{p}_r) + \dot{\mathbf{p}}_r \tag{17}$$

By Equation (4), Condition (14) and (15), $|\theta_c| = |\theta_e| \ll 1$, $|\dot{\theta}_c| = |\dot{\theta}_e| \ll 1$. Therefore T_e and $\dot{T}_e$ in Equation (17) can be considered as the identity matrix and null matrix respectively. Therefore,

$$\dot{\mathbf{p}}_c = A\,(\mathbf{p}_c - \mathbf{p}_r) + \dot{\mathbf{p}}_r$$

By substituting the previous equation by Equation (4) and condition (14), we obtain Equation (16). □

Equation (16) shows the behavior x_c is independent of y_c and θ_c is this small perturbation case. $1/K_x$ corresponds to the time constant of the exponential decay.

Corollary 1 When $x_c = \Delta x$ at $t = 0$,

$$x_c = V_r t + \Delta x\, e^{-K_x t} \tag{18}$$

By cancelling θ_e in Equation (16),

$$\ddot{y}_c + 2\,\zeta\,\xi\,\dot{y}_c + \xi^2 = 0 \tag{19}$$

where,

$$\zeta = \frac{K_\theta}{2\sqrt{K_y}} \text{ and } \xi = V_r \sqrt{K_y}$$

Corollary 2 The condition for critical damping is

$$K_\theta^2 = 4\,K_y \tag{20}$$

Corollary 3 In critical damping and if $y_c = \Delta y$ and $\theta_c = 0$ at $t = 0$,

$$\mathbf{p}_c = \begin{bmatrix} V_r t \\ \Delta y\,(1 + \xi t) e^{-\xi t} \\ -\Delta y\,\xi^2\, t\, e^{-\xi t} \end{bmatrix} \tag{21}$$

In this motion, the error ratio of $y_c/\Delta y$ is reduced to 9.2% when x_c becomes $4/\sqrt{K_y}$.

Simulation results on three distinct convergence characteristics are shown in Figure 4. Here, the robot's $\mathbf{p}_r$ and $\mathbf{p}_c$ were moving on the x axis to the positive direction, when y_r suddenly jumps up with $\Delta y = 5cm$ while continuing a parallel horizontal reference motion. The common parameters are $v_r = 30cm/sec$, $K_y = 6.4 \times 10^{-3}/cm$, $\xi = 2.4/sec$, and $K_x = 10/sec$. An over dumping case ($\zeta = 0.75$ and $K_\theta = 0.12/cm$), a critical dumping case ($\zeta = 1$ and $K_\theta = 0.16/cm$), and an oscillatory case ($\zeta = 1.25$ and $K_\theta = 0.20/cm$) are used. With the small Δy perturbation of 5cm, the result of simulation and analysis matches. In the following experiments and in our real implementation, we adopt the critical damping condition, $\zeta = 1$, since the convergence is fastest under non-oscillatory condition.

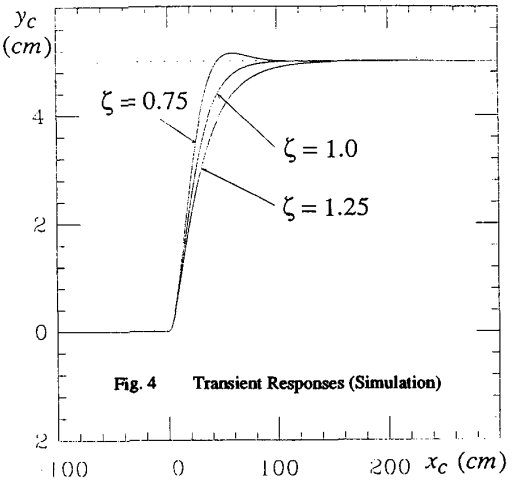

Fig. 4 Transient Responses (Simulation)

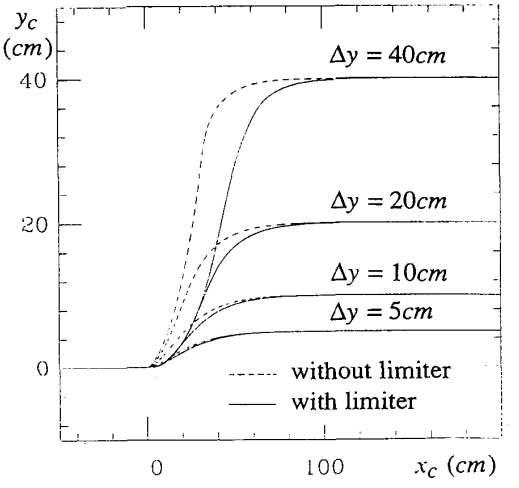

Fig. 5 Lateral Discontinuity

5. Velocity/Acceleration Limiting

For this tracking controller system, reference paths designated by $\mathbf{p}_r(t)$ and $\mathbf{q}_r(t)$ should satisfy the following conditions for "smoothness" (Preposition 2); (a) the path itself is continuous, (d) the path has tangent direction continuity, (c) the path curvature is continuous, (d) the derivative $\dot{v}_r$ is bounded, and (e) the derivative $\dot{\omega}_r$ is bounded (and hence, the derivative of curvature is also bounded). This curvature continuity requirement (c) is the reason why *clothoid curves*, *cubic spirals*, and *polar polynomials*

have been developed for vehicle path planning [5][13][14].

From a vehicle navigator's viewpoint, however, it is convenient if non-smooth paths are allowed to use. Although a path consisting of a line segment and a circular arc does not possess curvature continuity, that kind of paths are widely used [7][15][16][17]. In the MML language on the Yamabico-11 mobile robot, a function called *set_current(&p)* is provided to compensate the robot's positional error dynamically, and hence, it is frequently used in real-time navigation experiments [19]. However, if we allow these non-smooth paths, (i) either or both of the target velocities (v, ω) by Equation (8) might become too large to be attained by a real vehicle, and (ii) the linear/rotational acceleration might become too large causing the robot's slippage (Any slippage is a cause of a severe error in dead-reckoning). Therefore, in order to handle those non-smooth reference paths, we need some limiter for velocities and accelerations. We adopt a simple algorithm of limiting the target velocities (v, ω) by constants ($\hat{v}$, $\hat{\omega}$) and the target accelerations (a, α) by constants ($\hat{a}$, $\hat{\alpha}$), where $a = \dot{v}$ is a *linear target acceleration* and $\alpha = \dot{\omega}$ a *rotational target acceleration*. This modification is implemented in the box T in Figure 3.

Figure 5 shows simulation results for various values of Δy's with and without the velocity/acceleration limiter. Notice that the responses with a limiter are slower than that without a limiter. Hereafter, all simulations are done using the critical damping parameter set. Figure 6 shows simulation results for $\Delta\theta$ discontinuous jumps without limitation ($\Delta\theta = \pi/4$, $\pi/2$ and $3\pi/4$). Figure 7 shows simulation results for $\Delta\theta$ discontinuous jumps with velocity/acceleration limitation.

6. Implementation

The results presented in Sections 3, 4 and 5 were hardware independent. In this Section, we will describe how the theory was implemented on the robot Yamabico-11.

6.1. Determining Control Parameters

A larger K_x makes convergence faster and reduces a steady error x_e. However, it is not appropriate to have a time constant $1/K_x$ comparable to the sampling time of the robot's hardware. With a larger K_x, the control system tends to be oscillatory and instable even in its stop state (where $\mathbf{p}_r = constant$). An oscillation is observed at $K_x = 30/sec$, when time constant $1/K_x = 33ms$ is compatible to the robot's sampling time $T_s = 10ms$. Balancing these factors, $K_x = 10/sec$ was chosen.

Adopting the critical damping condition ($\zeta = 1$) in Section 4, we also to determine a value of ξ for appropriate response of current posture $\mathbf{p}_c$. A larger ξ makes convergence faster. However, a too large ξ demands the robot an excessive rotational velocity. We decided to adjust the parameters so that the robot will reduce the error $y_c/\Delta y$ into 9.2% during a 50cm run after a small perturbation of Δy. Therefore, by Corollaries 2 and 3, value $K_y = 6.4\times10^{-3}/cm^2$ and $K_\theta = 0.16/cm$ are determined. In this case, the time constant $1/\xi = 1/2.4sec$ is sufficiently larger than the sampling time $T_s = 10ms$.

With these K_x, K_y and K_θ, no oscillations were seen. The errors x_e and y_e at constant reference velocity of $30cm/sec$ are about $2mm$ and less than $1mm$ respectively.

6.2. Determining Maximum Velocity/Acceleration

The maximum linear velocity of the Yamabico-11 is known as 65cm/sec. We must consider the condition that even when the robot runs at $\hat{v}$ and rotates at $\hat{\omega}$ at the same time, the velocity of the outer wheel should not exceed that maximum velocity, 65cm/sec. Thus,

$$\hat{v} + \hat{\omega}(\frac{W}{2}) < 65cm/sec,$$

where W is the tread ($52.4cm$). Through this relation, we chose the maximum velocities as $(\hat{v}, \hat{\omega}) = (40cm/sec, 0.8rad/sec)$. We determine the values of $\hat{a}$ and $\hat{\alpha}$ by experiments with which the robot never slips: $(\hat{a}, \hat{\alpha}) = (50cm/sec^2, 5rad/sec)$.

6.3. Experimental Results

We conducted a few experiments to make sure that these values of K_y and K_θ are reasonable. Figure 8 shows experimental results with three distinct values of ζ, which corresponds to Figure 4. Figure 9 shows results on Δy, which corresponds to Figure 5. Figure 10 shows results on $\Delta\theta$, which corresponds to Figure 7. (As shown here, the results on the real vehicle are close to that of simulation with a velocity/acceleration limiter.) In Figures 8-10, the trajectories are plotted using the current posture $\mathbf{p}_c$ which is obtained by the vehicle's dead reckoning.

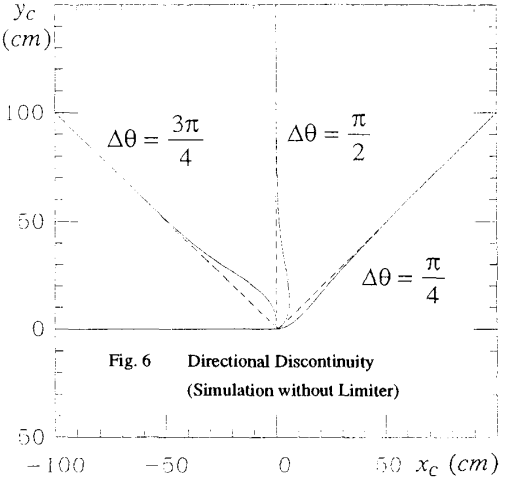

Fig. 6 Directional Discontinuity (Simulation without Limiter)

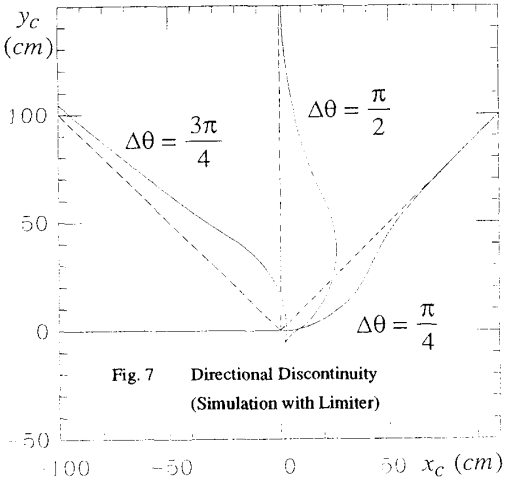

Fig. 7 Directional Discontinuity (Simulation with Limiter)

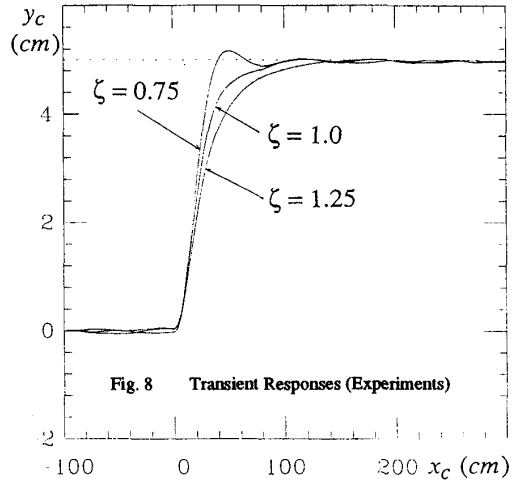

Fig. 8 Transient Responses (Experiments)

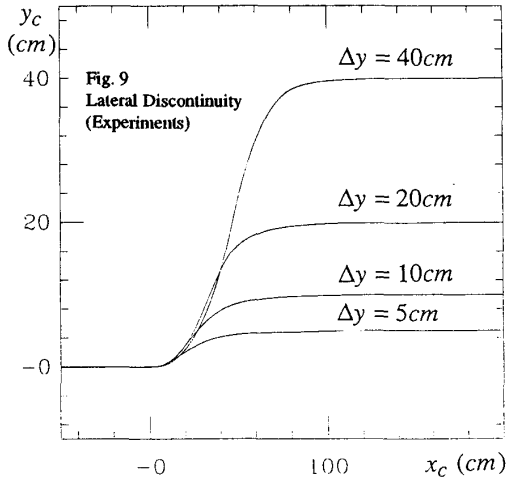

Fig. 9
Lateral Discontinuity
(Experiments)

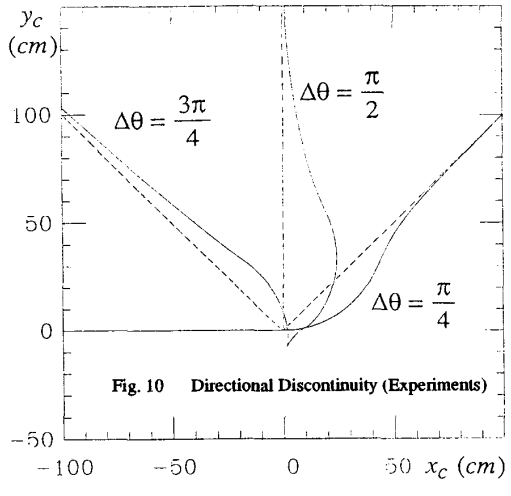

Fig. 10 Directional Discontinuity (Experiments)

Acknowledgement

The authors thank Dr. Shin'ichi Yuta, Michiyuki Shindo and Teijiro Kajiwara for the design and construction of Yamabico-11, and Amir Nilipour and Tony Lelm for their design and implementation of the earlier version of the Yamabico software system including the tracking controller. The authors also are thankful to Dr. Daniel Koditschek for his helpful comments on the stability theory.

References

[1] C. M. Wang, "Location Estimation and Uncertainty Analysis for Mobile Robots", Proc. IEEE International Conference on Robotics and Automation, pp. 1230-1235, 1988.

[2] T. Tsumura, N. Fujiwara, T. Shirakawa and M. Hashimoto, "An Experimental System for Automatic Guidance of Robot Vehicle, following the route stored in Memory," Proceeding 11th International Symposium on Industrial Robots, October 1981, pp. 187-193.

[3] Y. Kanayama and S. Yuta, "Vehicle Path Specification by a Sequence of Straight Lines", IEEE Journal of Robotics and Automation, vol. 4, no. 3, pp. 265-276, 1988.

[4] J. Crowley, "Asynchronous Control of Orientation and Displacement in a Robotic Vehicle", Proc. IEEE Conference on Robotics and Automation, pp. 1277-1282, 1989.

[5] Y. Kanayama and N. Miyake, "Trajectory Generation for Mobile Robots", Robotics Research, vol. 3, pp. 333-340, The MIT Press, 1986.

[6] S. Singh and D. H. Shin, "Position Based Path Tracking for Wheeled Mobile Robots", Proc. IEEE International Workshop on Intelligent Robots and Systems, in Tsukuba, Japan, pp. 386-391, September 1989.

[7] Y. Kanayama, A. Nilipour and C. Lelm, "A Locomotion Control Method for Autonomous Vehicles", Proc. IEEE Conference on Robotics and Automation, pp. 1315-1317, 1988.

[8] W. Nelson and I. Cox, "Local Path Control for an Autonomous Vehicles", Proc. IEEE Conference on Robotics and Automation, pp. 1504-1510, 1988.

[9] M. Vidyasagar, "Nonlinear Systems Analysis", Prentice-Hall Inc, Englewood Cliffs, N.J., 1978.

[10] F. Miyazaki, Y Masutani, C. Lelm and Y. Kanayama, "Precise Trajectory Following Control for Autonomous Vehicles", Proc. Annual Conf. of Institute of Systems, Control and Information Engineers, in Kyoto, Japan, May 1989.

[12] D. Koditschek, "Application of a New Lyapunov Function: Global Adaptive Inverse Dynamics for a Single Rigid Body", Repot of Center for Systems Science - 8806, Yale University, 1988.

[13] Y. Kanayama and B. Hartman, "Smooth Local Path Planning for Autonomous Vehicles", Proc. IEEE International Conference on Robotics and Automation, pp. 1265-1270, 1989.

[14] W. Nelson, "Continuous-Curvature Paths for Autonomous Vehicles", Proc. IEEE International Conference on Robotics and Automation, pp. 1260-1264, 1989.

[15] T. Lozano-Perez, and M. A. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Comm. ACM, vol. 22, No. 10, pp. 560-570, October 1979.

[16] K. Komoriya, S. Tachi and K. Tanie, "A Method for Autonomous Locomotion of Mobile Robots," Journal of Robotics Society of Japan, vol. 2, pp. 222-231, 1984.

[17] T. Hongo and H. Arakawa, G. Sugimoto, K. Tange, Y. Yamamoto, "An Automatic Guidance System of a Self-Controlled Vehicle - The Command System and the Control Algorithm ---," Proc. IECON, 1985.

[18] Y. Kanayama and T. Noguchi, "Locomotion Functions for a Mobile Robot Language", Proc. International Workshop on Advanced Robots and Intelligent Systems, pp. 542-549, September 1989.

[19] B. Hartman and Y. Kanayama, "Model and Sensor Based Precise Navigation by an Autonomous Mobile Robot", Proc. ICAR, 1989.