A Stable Tracking Control Method for an Autonomous Mobile Robot

Kanayama, Yutaka; Kimura, Yoshihiko; Miyazaki, Fumio; Noguchi, Tetsuo

http://hdl.handle.net/10945/40155

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.
A Stable Tracking Control Method for an Autonomous Mobile Robot

Yutaka Kanayama
Dept of Computer Science
Naval Postgraduate School
Monterey, CA 93943

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

Fumio Miyazaki
Toyama, 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 \((x, y, \theta)\) and reference velocities \((v, o, \omega)\). The major objective of this paper is to propose a control rule to find a reasonable target linear and rotational velocities \((v', o')\). Linearizing the system's differential equation is useful to decide parameters for critical damping 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 limiters. The control rule and limiting method proposed in this paper are robot-independent and hence can be applied to various kinds of mobile robots with a dead reckoning ability. This method was implemented on the autonomous mobile robot Yamabico-1. 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-1 are also presented.

Kanayama 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 computed 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 interpolated 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 \(\theta\) 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 \(o\)) for locomotion control, although vehicles have three degrees of freedom, \(x, y\) and \(\theta\) in its positioning. Another difficulty is in the non-linearity of the kinematic relation between \((v, o)\) and \((x, y, \theta)\). The use of a Liapunov function resolves these difficulties.

By linearizing the system's differential equation, we find a condition for critical damping, 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 omnidirectional) 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-II 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
possesses three degrees of freedom in its positioning which are represented by a posture.

\[ p = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \]  

where the heading direction \( \theta \) is taken counterclockwise from the \( x \)-axis. Let \( 0 \) denote a null posture \( (0, 0, 2\pi n)' \), where \( n \) is an integer. Since the robot has a locomotion capability in the plane, the posture \( p \) 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 \( x \) and \( y \) exist, \( \theta(t) \) is not an independent variable any more, because

\[ \theta(t) = \tan^{-1}\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \]  

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

\[ J = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ \sin \theta & 0 & \cos \theta \\ 0 & 0 & 1 \end{bmatrix} \]  

where \( q = (v, \omega)' \). This kinematics is common to all kinds of vehicles which are not omnidirectional. (For instance, an automobile, a bicycle, a vehicle with two parallel independent power wheels - power wheeled steering system, and a tricycle.)

2.2. Error Posture

In this control system, two postures are used: the reference posture \( p_r = (x_r, y_r, \theta_r)' \) and the current posture \( p_c = (x_c, y_c, \theta_c)' \). 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 \( p_e \), which is a transformation of the reference posture \( p_r \) in a local coordinate system with an origin of \( (x_c, y_c) \) and an \( x \)-axis in the direction of \( \theta \) \( \equiv \theta_r \) (Fig. 2). This is the "difference" between \( p_r \) and \( p_c \):

\[ p_e = J(p_c - p_r) \]  

If \( p_c = p_r \), the error posture \( p_e = 0 \). If \( p_c \) is ahead of \( p_r \) (the vehicle is behind of the goal), \( x_c > 0 \). For instance, if \( p_r = (3/2, 1, \pi/6)' \) and \( p_c = (\sqrt{3}/2, 1, 3\pi/4)' \), \( p_e = (\sqrt{3}/2, 1, \pi/12)' \) (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 \( p_r \) and reference velocities \( q_r = (v_r, \omega_r)' \), which are variables of time. The global output of the system is the current posture \( p_c \). The purpose of this tracking controller is to converge the error posture to 0. Let us describe each component in Figure 3 from left to right. The first component calculates an error posture from \( p_r \) and \( p_c \) using Equation (4). The second box is a control rule for the vehicle, which calculates a target velocities \( q = (v, \omega)' \) using the error posture \( p_e \) and the reference velocities \( q_r = (v_r, \omega_r)' \):

\[ q = q_r = J(p_r - p_c) \]  

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,
specifically, we assume the identity transformation:

\[ q_n = [v_n] = [v_n] = q \]  

(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 \( p_n \). The last box is for integration. Thus, only unknown component in this system is the control rule. Since the system's input \( p_n \) 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.**

\[
\begin{bmatrix}
\dot{x}_n \\
\dot{y}_n
\end{bmatrix} = f(x_n, y_n)
\]  

(7)

Proof. Using Equation (4) and an equality \( \dot{x}_n = \dot{y}_n = \dot{v}_n = 0 \), from Equation (3),

\[
\begin{align*}
\dot{x}_n &= (x_n - \dot{x}_n) \cos \theta + (y_n - \dot{y}_n) \sin \theta \\
\dot{y}_n &= -(x_n - \dot{x}_n) \sin \theta + (y_n - \dot{y}_n) \cos \theta
\end{align*}
\]

is rewritten as

\[
\begin{align*}
\dot{x}_n &= x_n - \dot{x}_n \cos \theta - y_n \sin \theta \\
\dot{y}_n &= x_n \sin \theta + y_n \cos \theta
\end{align*}
\]

Thus, \( \dot{x}_n = 0 \) and \( \dot{y}_n = 0 \). Substituting \( \dot{x}_n \) and \( \dot{y}_n \) by \( v(p_n, q_n) \) and \( r(p_n, q_n) \) respectively (cf. Equations (5) and (6)), we obtain the lemma.

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

\[
\begin{bmatrix}
\dot{x}_n \\
\dot{y}_n
\end{bmatrix} = f(x_n, y_n) = \begin{bmatrix}
v_n \cos \theta_n + K_x \dot{x}_n \\
v_n + r_n \sin \theta_n + K_y \dot{y}_n
\end{bmatrix}
\]  

(8)

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

**Lemma 2.**

\[
\begin{bmatrix}
p_x \\
p_y
\end{bmatrix} = f(x_n, y_n) = \begin{bmatrix}
(v_0 + r_n(K_x \dot{x}_n + K_a \sin \theta_n)) \dot{x}_n - K_x \dot{x}_n \\
(v_0 + r_n(K_x \dot{x}_n + K_a \sin \theta_n)) \dot{y}_n + K_a \dot{y}_n
\end{bmatrix}
\]  

(9)

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

**Proposition 1.** If we use the control rule (8), \( p_n = 0 \) is a stable equilibrium point if the reference velocity \( v_n > 0 \).

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

\[
V = \frac{1}{2} (v_n^2 + \dot{y}_n^2) + (1 - v_n \sin \theta_n) / K_y
\]  

(10)

Clearly, \( V \geq 0 \). If \( p_n = 0 \), \( V = 0 \). If \( p_n \neq 0 \), \( V > 0 \). Furthermore, by Lemma 2:

\[
\begin{align*}
\dot{V} &= \dot{x}_n \dot{v}_n + \dot{y}_n \dot{\dot{y}}_n + \dot{\theta}_n \dot{\theta}_n \\
&= (v_0 + r_n(K_x \dot{x}_n + K_a \sin \theta_n)) \dot{x}_n + (v_0 + r_n(K_x \dot{x}_n + K_a \sin \theta_n)) \dot{y}_n + \dot{\theta}_n \dot{\theta}_n
\end{align*}
\]

\[
= -K_a \dot{y}_n^2 - V_a \sin \theta_n \theta_n \leq 0
\]

Then, \( V \) becomes a Liapunov function.

The following proposition demonstrates that the uniformly asymptotically stability around \( p_n = 0 \) under some conditions:

**Proposition 2.** Assume that (a) \( v_n \) and \( \dot{v}_n \) are continuous, (b) \( v_0, \dot{v}_0, K_x \), and \( K_a \) are bounded, and (c) \( v_n \) and \( \dot{v}_n \) are sufficiently small. Under these conditions, \( p_n = 0 \) is uniformly asymptotically stable over \( [0, \infty) \).

Proof. By linearizing the differential Equation (9) around \( p_n = 0 \),

\[
\begin{bmatrix}
p_x \\
p_y
\end{bmatrix} = A \begin{bmatrix}
p_x \\
p_y
\end{bmatrix}
\]

(11)

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

\[
\begin{align*}
\lambda^3 + a_1 \lambda^2 + a_2 \lambda + a_3 &= 0
\end{align*}
\]

(12)

where

\[
\begin{align*}
a_1 &= -a_0 \dot{v}_n \cos \theta_n \\
a_2 &= -a_0 \dot{v}_n \sin \theta_n + a_1 \sin \theta_n \\
a_3 &= a_0 \v_n \cos \theta_n
\end{align*}
\]

(13)

The characteristic equation above is a cubic equation with real coefficients. Since all coefficients \( a_i \) are positive and \( a_1 a_2 - a_3 > 0 \), 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 \( K_a \). 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 at a constant velocity \( V_x \).
This condition is called the linear reference motion. In addition, we assume that:

\[ \left| \theta_e \right| < 1 \quad \text{and} \quad \left| \dot{\theta}_e \right| < 1 \]  

(15)

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

\[
\dot{p}_e = \begin{bmatrix}
\dot{x}_e \\
\dot{y}_e \\
\dot{\theta}_e
\end{bmatrix} = \begin{bmatrix}
-K_e & 0 & 0 \\
0 & V_e & 0 \\
0 & -V_e & 0
\end{bmatrix} \begin{bmatrix}
x_e - V_t \\
y_e \\
\theta_e
\end{bmatrix} + \begin{bmatrix}
V_t \\
0 \\
0
\end{bmatrix}
\]  

(16)

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

\[
\dot{p}_e = T_e^{-1} (A T_e - T_e X_p - p_e) + \dot{\theta}_e
\]  

By Equation (4), Conditions (14) and (15), \( \left| \theta_e \right| = \left| \theta_e \right| < 1 \), \( \left| \dot{\theta}_e \right| = \left| \dot{\theta}_e \right| < 1 \). Therefore, \( T_e \) and \( T_e^{-1} \) can be considered as the identity matrix and null matrix respectively. Therefore,

\[
p_e = A (p_e - \dot{p}_e) + \dot{\theta}_e
\]

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

Equation (16) shows the behavior \( x_e \) is independent of \( y_e \) and \( \theta_e \) in this small perturbation case. \( 1/e \) corresponds to the time constant of the exponential decay.

**Corollary 1**. When \( x_e = \Delta x \) at \( t = 0 \),

\[
x_e = V_t \Delta t + \Delta x e^{-\xi t}
\]  

(18)

By cancelling \( \dot{\theta}_e \) in Equation (16),

\[
y_e = V_t \Delta t + \xi t^2 + \xi^2 t^3 = 0
\]  

(19)

where,

\[
\xi = \frac{K_g}{2 \sqrt{K_p}} \quad \text{and} \quad \zeta = V_t \sqrt{K_g}
\]

**Corollary 2.** The condition for critical damping is

\[
K_g^2 = 4K_p
\]  

(20)

**Corollary 3.** In critical damping and if \( y_e = \Delta y \) and \( \theta_e = 0 \) at \( t = 0 \),

\[
\psi_e = \begin{bmatrix}
V_t t \\
\Delta y (1 + \frac{\xi}{\zeta}) e^{-\xi t} \\
-\Delta y \xi^2 t^2 e^{-\xi t}
\end{bmatrix}
\]  

(21)

In this motion, the error ratio of \( y_e / \Delta y \) is reduced to 9.2% when \( x_e \) becomes \( 4/\sqrt{K_p} \).

Simulation results on three distinct convergence characteristics are shown in Figure 4. Here, the robot's \( p_e \) and \( p_r \) were moving on the x-axis to the positive direction, when \( y_e \) suddenly jumps up with \( \Delta y = 15 \) while continuing a parallel horizontal reference motion. The common parameters are \( v_e = 30 \text{cm/sec}, K_p = 6.4 \times 10^{-9} \text{cm}, \zeta = 2.4 \text{sec}, \) and \( K_g = 10 \text{sec}^{-1} \).

An over damping case \((\zeta = 0.75 \text{ and } K_g = 0.12 \text{ cm})\), a critical damping case \((\zeta = 1 \text{ and } K_g = 0.16 \text{ cm})\), and an oscillatory case \((\zeta = 1.25 \text{ and } K_g = 0.20 \text{ cm})\) are used. With the small \( \Delta y \) perturbation of 5 cm, 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.

5. Velocity/Acceleration Limiting

For this tracking controller system, reference paths designated by \( p_e(t) \) and \( q_e(t) \) should satisfy the following conditions for "smoothness" (Proposition 2): (a) the path itself is continuous, (d) the layer has tangent direction continuity, (c) the path curvature is continuous, (d) the derivative \( y_e \) is bounded, and (e) the derivative \( \dot{y}_e \) is bounded (and hence, the derivative of curvature is also bounded). This curvature continuity requirement is the reason why cycloidal curves, cubic splines, and polar polynomials are used.
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 be used. 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 [18]. However, if we allow these non-smooth paths, (i) either of both of the target velocities (\(v_a, a\)) 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 these non-smooth reference paths, we need some limiter for velocities and accelerations. We adopt a simple algorithm of limiting the target velocities (\(v_a, a\)) by constants (\(v_a^*, a^*\)) and the target accelerations (\(a_v, a_r\)) by constants (\(a_v^*, a_r^*\)).

\[
\begin{align*}
&v_a = \min(v_a^*, \max(v_a, 0)) \\
&a_r = \min(a_r^*, \max(a_r, 0))
\end{align*}
\]

This modification is implemented in the box \(T\) in Figure 3.

Figure 5 shows simulation results for various values of \(\Delta v, a\)'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 parameters set. Figure 6 shows simulation results for \(\Delta v\) discontinuous jumps without limitation (\(\Delta v = \pi/4, \pi/2, \pi\)) and \(\Delta x = 10\text{cm}\). Figure 7 shows simulation results for \(\Delta v\) 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 target \(K_v\) makes convergence faster and reduces a steady error \(x_c\), However, it is not appropriate to have a time constant \(1/K_v\) comparable to the sampling time of the robot's hardware. With a larger \(K_v\), the control system tends to be oscillatory and unstable even in its stop state (where \(p_c = \text{constant}\)). An oscillation is observed at \(K_v = 30\text{sec}\), when time constant \(1/K_v = 33\text{ms}\) is comparable to the robot's sampling time \(T_s = 10\text{ms}\). Balancing these factors, \(K_v = 10\text{sec}\) was chosen.

Adopting the critical damping condition (\(\xi = 1\)) in Section 4, we also determine a value of \(K_v\) for appropriate response of current posture \(p_c\). A larger \(\xi\) makes convergence faster. However, a too large \(\xi\) demands the robot an excessive rotational velocity. We decided to adjust the parameters so that the robot will reduce the error \(p_c/\Delta v\) into 9.2% after a small perturbation of \(\Delta v\). Therefore, by Corollaries 2 and 3, value \(K_v = 6.4 \times 10^3\text{cm}^2/\text{sec}^2\) and \(X_d = 0.16\text{cm}^3/\text{sec}\) are determined. In this case, the time constant \(1/\xi = 1/2\text{sec}\) is sufficiently larger than the sampling time \(T_s = 10\text{ms}\).

With these \(K_v, K_x\), and \(K_a\), no oscillations were seen. The errors \(x\) and \(x_c\) 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 \(v\) and rotates at \(a\) at the same time, the velocity of the outer wheel should not exceed that maximum velocity, 65cm/sec. Thus,

\[
v + a \pi \left( \frac{W}{2} \right)^2 < 65\text{cm/sec}.
\]

where \(W\) is the tread (52.4cm). Through this relation, we choose the maximum velocity as \((v, a) = (40\text{cm/sec}, 0.8\text{rad/sec})\). We determine the values of \(\dot{v}\) and \(\dot{a}\) by experiments with which the robot never slips: \((\dot{v}, \dot{a}) = (50\text{cm/sec}, 0.8\text{rad/sec})\).

6.3. Experimental Results

We conducted several experiments to make sure that these values of \(K_v\), \(K_x\), and \(K_a\) are reasonable. Figure 8 shows experimental results with three distinct values of \(\Delta y\), which corresponds to Figure 4. Figure 9 shows results on \(\Delta y\), which corresponds to Figure 5. Figure 10 shows results on \(\Delta y\), 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 \(p_c\), which is obtained by the vehicle's dead reckoning.
Acknowledgement

The authors thank Dr. Shin'ichi Yuta, Michiyuki Shinbo and Teijiro Kajiwara for the design and construction of Yamabico-I1, and Amir Nili- poor and Tony Lohr 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