Stability and Four-Posture Control for Nonholonomic Mobile Robots

Hyun-Sik Shim and Yoon-Gyeoung Sung

***Abstract*—In this paper, a four-posture control for nonholonomic mobile robots is proposed. Based upon posture velocity error dynamics, the empirical kinematic motion requirements of a mobile robot are proposed, and the four-posture control is designed in order to generate the robustly required moving trajectory with posture error reduction. The controller is designed with four possible moving directions of linear and angular velocities. The robustness of the controller is confirmed using Lyapunov stability theory. The proposed controller is experimentally demonstrated under high velocity and acceleration conditions with different control parameters. It is shown that the four-posture control algorithm is effective and feasible, even if an unsophisticated velocity controller is used as the deriving unit of a mobile robot.**

***Index Terms*—Lyapunov stability theory, nonholonomic mobile robot, tracking control.**

I. Introduction

It is known that the stabilization of wheeled mobile robots with nonholonomic constraints to an equilibrium state is quite difficult. As shown by Brockett [1], nonholonomic systems cannot be stabilized with smooth state feedback. The stabilization problem for driftless systems represents a challenge for nonlinear control theory, because the linearization of the system is not controllable. Therefore, one is forced to rely on the use of strongly nonlinear techniques to stabilize the system. Results in asymptotic stability typically rely on the use of discontinuous feedback, time-varying feedback, or a combination of the two. Even though a variety of control algorithms [2]–[4] for the nonholonomic mobile robot have been designed for the nonholonomic robot, no algorithm allows fast, consecutive performance in realistic applications. For the realistic implementations, it is difficult to get good performance due to the limitations of motor output, velocity controller performance, system uncertainty, etc.

Kanayama and Yuta [5] suggested a method which uses a straight line reference for the robot's locomotion, instead of a sequence of points. Crowley [6] developed a locomotion control system with a three-layered structure, which employed the concept of a "virtual vehicle." The concept of virtual structure is useful to construct a control system, which is independent of the robot. In its command system, independent control of linear and rotational motions is possible so that smooth curve trajectories can be obtained [7]. Aicardi *et al.* [9] presented a control algorithm, which used polar coordinates. Its stability was proven using the Lyapunov stability theory. For convenient analysis and motion control design, the algorithms in [8] and [9] were reported with numerical simulations. Recently, Yang and Kim [10] presented an algorithm with a sliding mode control with the inclusion of robot dynamics. These algorithms were evaluated under limited conditions rather than by using practical circumstances. Astolfi [11] presented a control mechanism with an exponential stabilization of the kinematic and dynamic model of a simple wheeled mobile robot by using a discontinous, bounded, time invariant, state feedback control law. He evaluated the control mechanism under modeling error and noisy measurement conditions. However, it was found necessary to evaluate Astolfi's result in the case of high velocity and acceleration, due to the zig-zag motion of a mobile robot, caused by output noise. Closkey and Murray [12] presented an algorithm to improve the convergence rate of driftless systems by using homogeneous feedback.

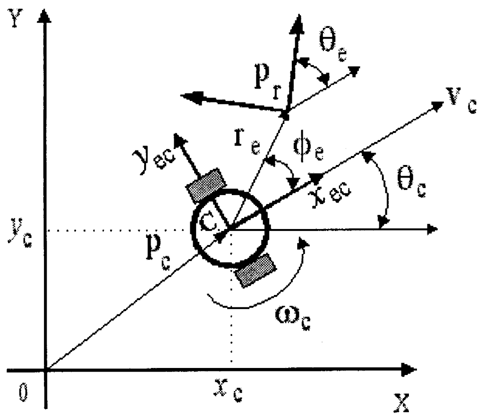

Fig. 1. Coordinate system of the mobile robot (subscript "c" refers to the "current state").

In this paper, we develop a controller by considering robot kinematics to largely reduce the amount of time required to arrive at a final posture. Contrary to previous control design approaches, the proposed algorithm is designed with given specifications by considering the robot's desired trajectory and the physical limitations of the actuators. The effectiveness and feasibility are then demonstrated in a practical sense through experiments.

The remainder of the paper is organized as follows. In Section II, the notations used and the kinematic model are defined. In Section III, a control law for motion control is designed. In Section IV, experimental results illustrate the effectiveness of the scheme. Finally, the conclusion is presented in Section V. In the Appendix, the stability of motion control is proved.

II. Kinematics of the Mobile Robot

Consider a mobile robot located on a two-dimenaional (2-D) plane defined by a global Cartesian coordinate system. Global coordinates $X - Y$ and local coordinates $x - y$ attached to the center of the robot body are shown in Fig. 1. The robot has two degrees of freedom (DOFs) of gross motion and three degrees of posture

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

where the heading direction θ is taken counterclockwise from the X axis to the x axis. The angle θ indicates the orientation of the robot or of the wheels. The set $(x\ y)$ represents the robot position. In Fig. 1, the vector $(x_c\ y_c\ \theta_c)^T$ corresponds to the current posture of the robot.

Manuscript received March 27, 2002; revised October 23, 2002. This paper was recommended for publication by Associate Editor E. Colgate and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by research funds from Chosun University, Gwangju, South Korea, 2003.

H.-S. Shim is with Road Traffic Information System (ROTIS) Inc., Seoul 133-110, South Korea.

Y.-G. Sung is with the Department of Mechanical Engineering, Chosun University, Gwangju 501-759, South Korea (e-mail for any correspondence: sungyg@chosun.ac.kr).

Digital Object Identifier 10.1109/TRA.2003.819730

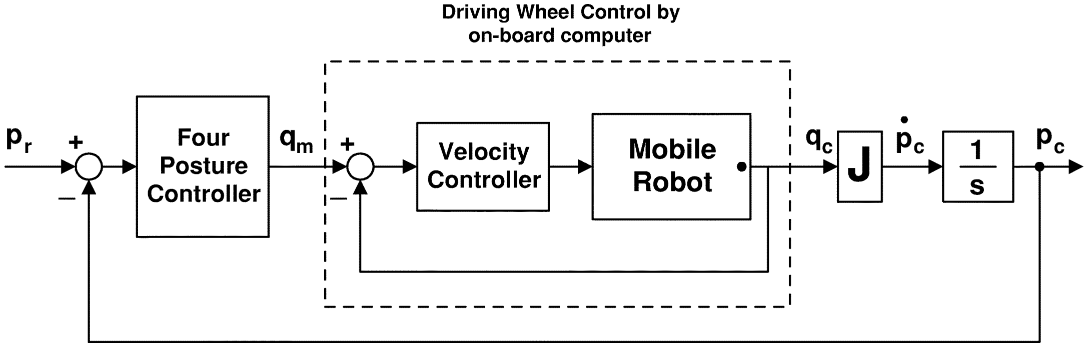

Fig. 2. Control diagram with four posture and velocity controllers [Jacobian matrix $\mathbf{J}$ of (2)].

Robot motion is controlled by its linear velocity v and angular velocity ω, which are functions of time. Robot kinematics are defined by the Jacobian matrix $J(\theta)$

$$\dot{\mathbf{p}} = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{Bmatrix} = J(\theta)\mathbf{q} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \mathbf{q} \tag{2}$$

where $\mathbf{q} = (v, \omega)^T$.

In terms of the mobile robot postures, there are two postures, i.e., the *reference posture* $\mathbf{p}_r = (x_r\ y_r\ \theta_r)^T$, which is the target posture, and the *current posture* $\mathbf{p}_c = (x_c\ y_c\ \theta_c)^T$. The robot-based error posture is expressed in polar coordinates $(r_e, \phi_e, \theta_e)^T$, which is defined by the posture error between $\mathbf{p}_r$ and $\mathbf{p}_c$ as

$$\mathbf{p}_{ec} = \begin{Bmatrix} x_{ec} \\ y_{ec} \\ \theta_{ec} \end{Bmatrix} = \begin{Bmatrix} r_e \cos\phi_e \\ r_e \sin\phi_e \\ \theta_e \end{Bmatrix} \tag{3}$$

where $r_e > 0$ is the distance error, $\theta_e = \theta_r - \theta_c$ is the orientation error with respect to $\mathbf{p}_c$, and ϕ_e is the angle measured from the robot's principle axis to the distance error vector r_e. The velocity of (3) in polar coordinates is expressed as

$$\begin{Bmatrix} \dot{r}_e \\ \dot{\phi}_e \\ \dot{\theta}_e \end{Bmatrix} = \begin{Bmatrix} -v_c \cos\phi_e \\ -\omega_c + \frac{v_c}{r_e}\sin\phi_e \\ -\omega_c \end{Bmatrix}. \tag{4}$$

Equation (4) will be used to design a four-posture controller in the next section.

III. Four-Posture Control

For the control element, based on robot kinematics, the control structure is composed of a posture controller and a velocity controller, as shown in Fig. 2. If the reference posture $\mathbf{p}_r$ is given, the controller commands the mobile robot to approach the reference posture. The posture controller makes the robot proceed to the reference posture through a velocity command $\mathbf{q}_m = (v_m, \omega_m)^T$. The velocity controller then makes the robot follow the command $\mathbf{q}_m$. If the velocity controller is designed with consideration of robot dynamics, friction, unknown disturbances, and motor saturation, etc., it can be assumed that the command velocity becomes the robot's velocity, $\mathbf{q}_m = \mathbf{q}_c$. Finally, $\mathbf{p}_c$ is obtained by using (2) and the integration of time. In this paper, a robust four-posture controller is presented under the assumption that the velocity controller is designed using control schemes [13], [14].

A. Motion Requirements

The design specifications, based on posture velocity error dynamics (4), following a moving trajectory or a velocity command, are given as follows:

1) if the distance to the reference posture is large, then movement is fast; speed is reduced on approach;
2) the robot should take the least amount of time to arrive at a final posture;
3) if the angle error ϕ_e is large, the robot's linear velocity should be reduced to minimize the distance moved;
4) if the reference posture is too far moved, the robot needs to first reduce the distance error;
5) the robot should meet the reference posture immediately.

These specifications require that the posture controller generates an optimal moving trajectory with minimum posture error. The posture controller is designed, using the vector $(r_e\ \phi_e\ \theta_e)^T$, and the velocity command for the robot is produced, based on the above specifications. In addition, the stability of the four-posture controller is expressed in the Appendix.

To meet the first and second specifications, a linear velocity controller with $v_m = k_r r_e$ is considered. This equation is similar to the equation proposed by Aicardi *et al.* [9]. When the distance is very large, the velocity commanded will be high, and the robot's velocity will be saturated. Instead of r_e, a new variable r_e' is introduced to consider such a situation

$$r_e' = \begin{cases} \frac{\pi}{2}, & r_e \geq R_m \\ \frac{\pi r_e}{2R_m}, & \text{otherwise.} \end{cases}$$

A "sine" relationship was chosen instead of a linear one, so that the velocity becomes higher, even with a smaller r_e

$$v_m = k_r \sin r_e'.$$

For the third specification, the angle error ϕ_e is used to control the linear velocity. To apply an inverse proportional relationship for ϕ_e, the following equation was employed:

$$v_m = k_r \sin r_e' \cos\phi_e.$$

The following variables are defined as limitations:

$$d = \begin{cases} 1 & -\Phi_m \leq \phi_e \leq \Phi_m \\ -1 & \phi_e < -\pi + \Phi_m \text{ or } \phi_e > \pi - \Phi_m \\ 0 & \text{otherwise} \end{cases}$$

$$\phi_e' = \begin{cases} \phi_e - \pi & \phi_e > \frac{\pi}{2} \\ \phi_e + \pi & \phi_e < -\frac{\pi}{2} \\ \phi_e & \text{otherwise} \end{cases}$$

$$\theta_e' = \begin{cases} \theta_e - \pi & \theta_e > \frac{\pi}{2} \\ \theta_e + \pi & \theta_e < -\frac{\pi}{2} \\ \theta_e & \text{otherwise} \end{cases}$$

where Φ_m is an arbitrary constant between 0 and $\pi/2$. By introducing d, ϕ_e', and θ_e', it is possible to make the robot move back and forth,

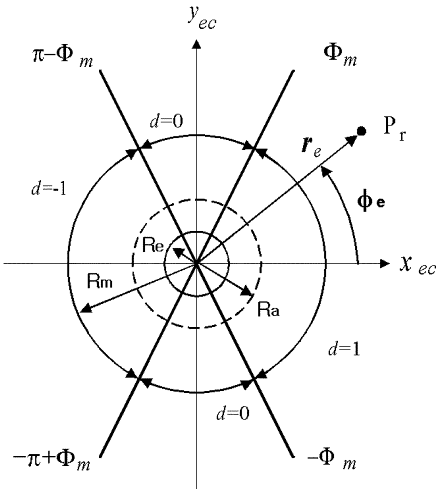

Fig. 3. Control variable assignment for the robot's location.

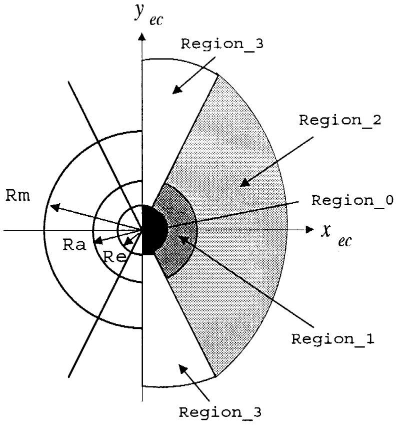

Fig. 4. Four classifications to r_e and ϕ_e. Region_0 $(r_e \leq R_e)$. Region_1 $(-\Phi_m \leq \phi_e \leq \Phi_m$ and $R_e \leq r_e \leq R_a)$. Region_2 $(-\Phi_m \leq \phi_e \leq \Phi_m$ and $r_e \geq R_a)$. Region_3 $(\Phi_m < \phi_e \leq \pi/2, -\pi/2 \leq \phi_e < -\Phi_m$ and $r_e > R_e)$.

because the front and rear sides of the mobile robot are not distinguishable. Thus, the linear velocity controller law, with new assigned variables, as shown in Fig. 3, is proposed to be

$$v_m = k_r d \sin r_e' \cos \phi_e' \tag{5}$$

where k_r is a positive constant with units of velocity, and d refers to the direction of the robot, so that the robot moves in a forward direction when $d = 1$, and a backward direction when $d = -1$. Because there is no difference between the front and rear of the robot, only the regions $r_e \geq 0$, $-\pi/2 \leq \phi_e \leq \pi/2$, and $-\pi/2 \leq \theta_e \leq \pi/2$ are considered. For each of the distance errors r_e and angle errors ϕ_e, four regions are classified, as noted in Fig. 4.

In order to regulate the angle and orientation errors, the angular velocity is discussed in detail. Based on the design concept above, a relationship equation can be written as

$$\omega_m = k_\theta \left(\sin \phi_e' + \sin \theta_e'\right)$$

where k_θ is a positive constant with units of angular velocity. When the reference posture is some distance away, the angular error ϕ_e should be considered to meet the fourth element of the specification. As the robot is approaching the reference posture, the orientation error θ_e is an important aspect of meeting the fourth specification. Therefore, the angular velocity control law is proposed as

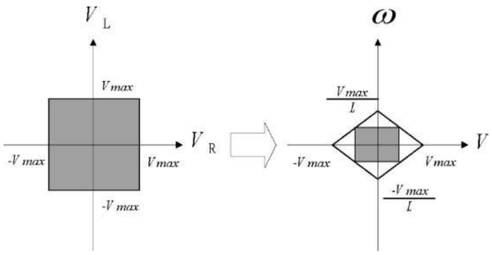

Fig. 5. Usable velocity area of the wheel and robot velocities.

$$\omega_m = k_\theta \left(\epsilon \sin \phi_e' + (1-\epsilon) \sin \theta_e'\right) \tag{6}$$

where variable ϵ is defined as

$$\epsilon = \begin{cases} 0 & r_e \leq R_e \\ 0.5 - \alpha \operatorname{sgn}(\sin \theta_e') \operatorname{sgn}(\sin \theta_e' + \sin \phi_e') & R_e < r_e < R_a \\ 1 & \text{otherwise} \end{cases}$$

where $0 < R_e \leq R_a \leq R_m$, $0 < \alpha < 0.5$, and the function $\operatorname{sgn}(x)$ is 1, if x is positive, and -1, otherwise.

The control variables for the proposed control law are k_r, k_θ, R_m, R_e, R_a, and Φ_m. Various trajectories can be generated based on their values. However, the trajectories do not differ as much as those of [8] and [9]. This is possible because the control laws for v_m and ω_m are designed on the basis of specific design specifications.

The variables, k_r and k_θ, are limited by the maximum velocity of the robot. However, the values of k_r and k_θ should be higher to meet the second specification. To tackle this problem, the robot's linear velocity v and angular velocity ω should be limited within the shadow areas shown in Fig. 5, because a robot's wheel velocity is limited by its maximum velocity $V_{\max}$. The robot should be controlled by the values within this area. If a high linear velocity v_m is given, the applied angular velocity ω_c should be limited in the calculation to a smaller value than the control command ω_m. Therefore, k_r and k_θ must be limited to

$$k_r + L k_\theta \leq V_{\max} \tag{7}$$

where L is the radius of the robot body.

For the implementation of motion control, several control parameters such as k_r, k_θ, R_m, R_e, R_a, and Φ_m are determined to get the desired robot motion trajectory. The following are descriptions of these parameters.

R_m is a deceleration factor for the linear velocity. If this is large, the velocity is reduced when the robot is too far away. A high value of this reduces slippage and makes the robot more stable than a smaller R_m. The robot will, however, take more time to reach the target point.

Φ_m should be decided because it is related to the moving trajectory. If it is close to 0°, the robot motion will turn around or go straight. However, if it is close to $\Phi_m = 90°$, the moving distance will be large. Therefore, Φ_m is an important factor which is related to the allowable error range (maximum R_e).

R_a is the radius of the circular area in which the control law has all three error terms. It can generate similar trajectories with smaller differences, according to Φ_m. If R_a has a small value, the trajectory is close to a straight line and the orientation error is reduced at the reference posture. If it has a large value, the trajectory is smooth and it may take a longer time for the robot to get to the target point.

R_e takes an allowable value based on the system performance, and its maximum value depends on Φ_m. If a small value is used, the robot arrival time will be delayed.

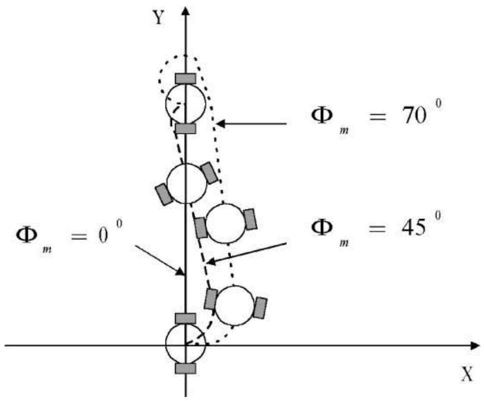

Fig. 6. Trajectory variation to Φ_m.

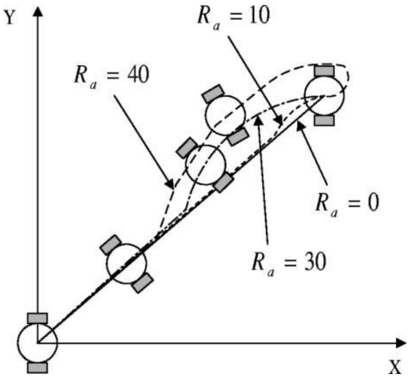

Fig. 7. Trajectory variation to R_a.

Figs. 6 and 7 show the analytical variations in motion trajectory according to the different parameters R_a and Φ_m. The effect of several parameters upon robot motion trajectory is evaluated in an experiment to reduce the arrival time at a final posture.

IV. Experimental Evaluation

A proportional-integral-derivative (PID) velocity controller for a robot is used to demonstrate the effectiveness and applicability of the proposed four-posture control law in Fig. 8. The robot, sized 7.5 cm $\times$ 7.5 cm $\times$ 7.5 cm, is designed for robot soccer games, and has an 80C196KC microprocessor running at 20 MHz, and two wheels driven by two DC motors with two LM629 motion controllers. The LM629 chip is used as a velocity controller, and receives only a reference velocity as the input signal. It then implicitly generates the required torque, according to the reference velocity, using a L298 driver chip. The maximum speed and acceleration of the robot are about 150 cm/s and 350 cm/s^2, respectively.

The sampling time of the four-posture control is 33 ms, depending on the vision processing time. The sampling time of the velocity controller is 410 μs. The command velocities (v, ω) from the four-posture controller are used to obtain the right and left wheel velocities (V_R, V_L) as

$$V_R = v + L\omega$$
$$V_L = v - L\omega$$

where L is the radius of the robot body. The host computer transmits the wheel velocities to the motion controller of the robot. However, accurate motion information is not provided to the robot system, because the command velocity of the motor is updated by 1 cm/s, and the locating accuracy of the robot's vision system is limited to 1 cm $\sim$ 2 cm.

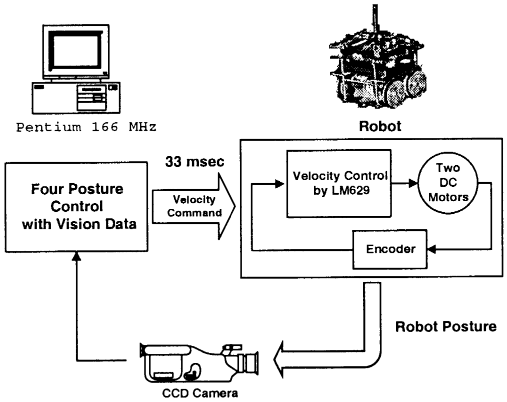

Fig. 8. Control block diagram of a vision-based mobile robot.

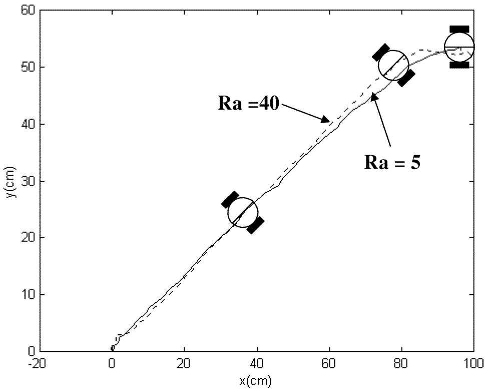

Fig. 9. Robot trajectories to the values of R_a ($k_r = 60, \Phi_m = 70°, R_m = 20, k_\theta = 10, p_r = (112, 72, 0°)$).

Therefore, the proposed algorithm will be demonstrated under a nonideal situation with the built-in PID velocity controlled by the LM629.

In this paper, R_e is chosen to be 2 cm, due to the locating accuracy of the vision system. This is a suitable choice, because it is greater than the maximum error of 1.607 cm when $\Phi_m = 70°$. The reference posture is given as (100, 50, 0°). Fig. 9 shows the robot trajectories, according to the values of R_a. The smaller the value of R_a is, the faster the robot can move to its reference posture. The figure shows that the orientation error is substantially reduced when the robot is close to the reference posture. Fig. 10 shows the robot trajectories, according to R_m. There is little difference between the two trajectories shown in Fig. 10, so that the robot is maneuvered along the reference posture. However, it takes 3.465 s for $R_m = 10$ and 6.1 s for $R_m = 50$ to produce the reference posture.

Fig. 11 shows the robot trajectories for various values of k_r. k_r refers to the maximum linear velocity of the robot with unit cm/s. If the velocity is higher than $k_r = 30$, the robot will quickly arrive at the reference posture. However, if the velocity is saturated with $k_r = 80$, the trajectory is slightly deviated from the design specifications. From Figs. 12-14, the robot movements with regard to time are shown toward the reference posture. It takes about 6 s for $k_r = 30$ and about 3 s for $k_r = 80$ to produce the reference posture. $k_r = 30$ produces more switching movements than $k_r = 80$ does, as shown in Figs. 13 and 14.

In Fig. 15, the robot trajectories are shown through the seven target points, which are located at (60, 47, 0°), (115, 70, 45°), (120, 48, 0°), $(113, 17, -45°)$, $(66, 13, 0°)$, $(13, 20, 0°)$ and $(16, 80, 90°)$ from "Start" to "Target 6." Based on the above experimental selections of control parameters, the proposed controller is experimentally evaluated. The

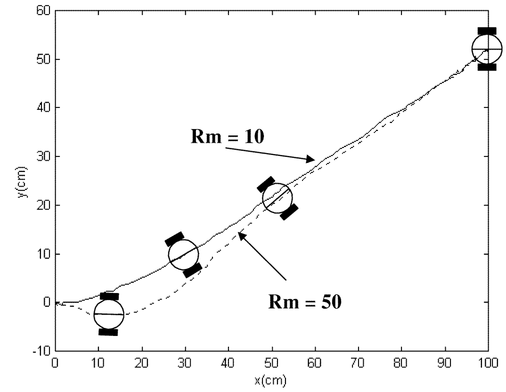

Fig. 10. Robot trajectories to the values of R_m ($k_r = 60, \Phi_m = 70°, R_a = 10, k_\theta = 10$).

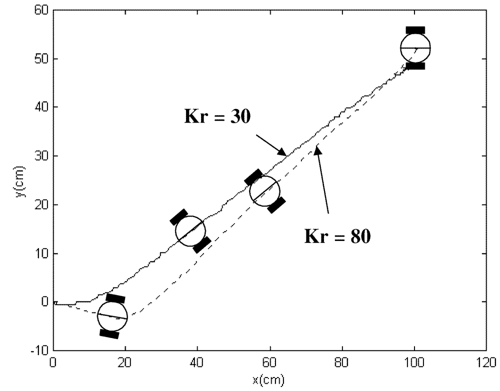

Fig. 11. Robot trajectories for the values of k_r ($R_a = 10, \Phi_m = 70°, R_m = 20, k_\theta = 10$).

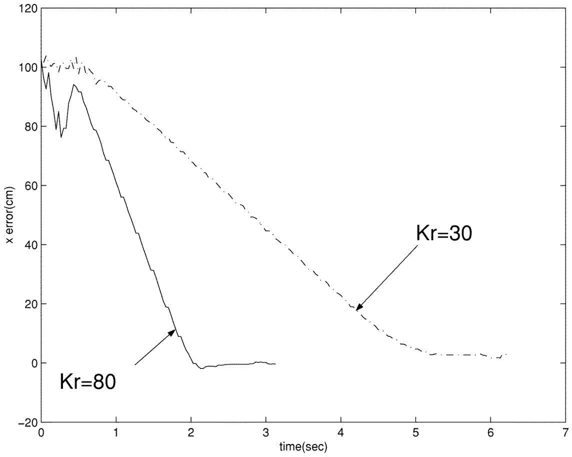

Fig. 12. x_{ec} with respect to k_r ($R_a = 10, \Phi_m = 70°, R_m = 20, k_\theta = 10$).

four-posture control can generate a usable motion trajectory by quickly arriving at specified target points.

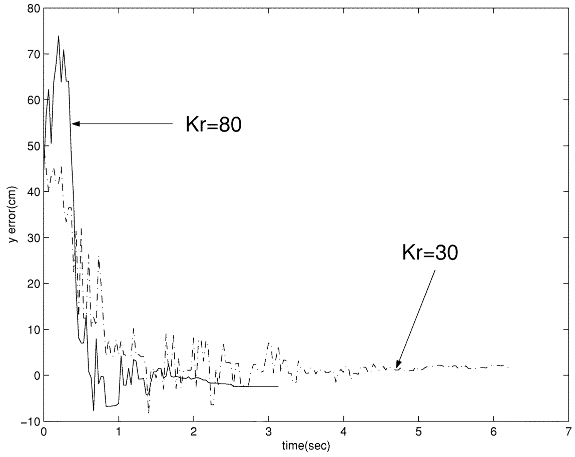

Fig. 13. y_{ec} with respect to k_r ($R_a = 10, \Phi_m = 70°, R_m = 20, k_\theta = 10$).

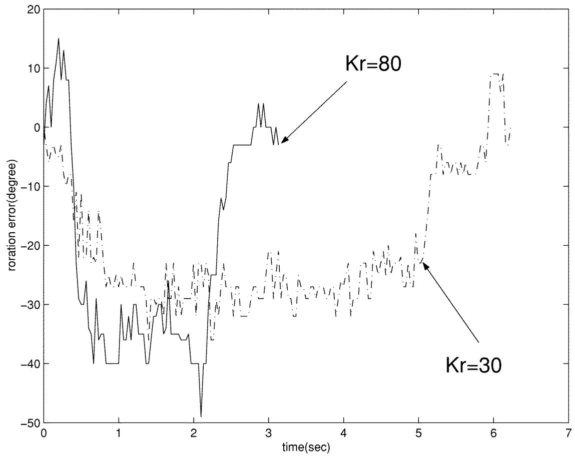

Fig. 14. θ_{ec} with respect to k_r ($R_a = 10, \Phi_m = 70°, R_m = 20, k_\theta = 10$).

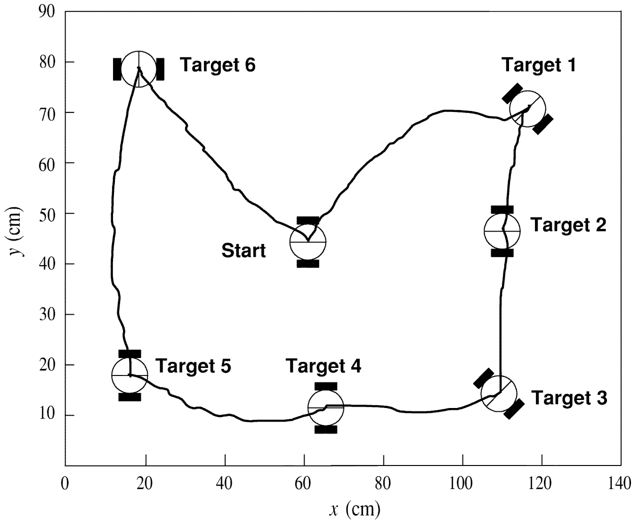

Fig. 15. Consecutive robot trajectories through the seven target points ($k_r = 60, R_a = 10, \Phi_m = 70°, R_m = 20, k_\theta = 10$).

V. Conclusions

In this paper, a four-posture control for nonholonomic mobile robots is proposed. Based on the given kinematic design specifications for the desired robot motion trajectory, a controller was developed with four possible moving regions for the mobile robot for linear and angular velocities. The robustness of the controller was verified by using the Lyapunov stability theory. The proposed control was demonstrated experimentally at high velocity and under high acceleration conditions with different selections of control parameters. Even if there exist velocity and acceleration limits in terms of the motor specifications, it is shown that the four-posture control algorithm is effective and feasible.

APPENDIX

PROOF OF CONTROL STABILITY

The following theorem represents the stability of four-posture control based on Lyapunov stability theory according to the robot's error ranges.

Theorem 1: Consider the kinematics of a mobile robot given by (2). The four-posture control laws in (5) and (6) drive the robot to converge to an allowable error boundary.

Proof: Choose a Lyapunov function candidate as follows:

$$V(r_e, \phi_e, \theta_e) = \frac{1}{2}r_e^2 + (1 - \cos\phi_e) + (1 - \cos\theta_e) \tag{8}$$

where $-\pi/2 \le \phi_e \le \pi/2$ and $-\pi/2 \le \theta_e \le \pi/2$. Substituting (5) and (6) for $\mathbf{q}_c$ with the assumption that $\mathbf{q}_m = \mathbf{q}_c$, the following equation is derived by differentiating with respect to time:

$$\begin{aligned}\dot{V}(r_e, \phi_e, \theta_e) &= \dot{r}_e r_e + \dot{\phi}_e \sin\phi_e + \dot{\theta}_e \sin\theta_e \\ &= -r_e v_c \cos\phi_e + \frac{\sin^2\phi_e}{r_e} v_c \\ &\quad - \omega_c \sin\phi_e - \omega_c \sin\theta_e \\ &= -\frac{k_r d \sin r_e'}{r_e} \cos\phi_e \left(r_e^2 \cos\phi_e - \sin^2\phi_e\right) \\ &\quad - k_\theta \sin^2\phi_e \epsilon - k_\theta \sin^2\theta_e (1 - \epsilon) \\ &\quad - k_\theta \sin\phi_e \sin\theta_e. \end{aligned} \tag{9}$$

Case 1: In the case of Region_3, $\Phi_m < \phi_e \le \pi/2, -\pi/2 \le \phi_e < -\Phi_m$, and $r_e > R_e$.

In Region_3, since $d = 0$ and $\epsilon = 1$, the control commands result in $v_m = 0$ and $\omega_m = k_\theta \epsilon \sin\phi_e$. Therefore, the derivative of the Lyapunov function is written as a function of ϕ_e and the last term of (9) is eliminated

$$\dot{V}(\phi_e) = -k_\theta \sin^2\phi_e \le 0.$$

As a result, the robot in Region_3 will move to another region.

Case 2: In the case of Region_2, $-\Phi_m \le \phi_e \le \Phi_m$ and $r_e \ge R_a$.

Since $d = 1$ and $\epsilon = 1$ in Region_2, the controller commands are functions of r_e and ϕ_e. Therefore, the derivative of the Lyapunov function is given as

$$\begin{aligned}&\dot{V}(r_e, \phi_e) \\ &\quad = -\frac{k_r \sin r_e'}{r_e} \cos\phi_e \left(r_e^2 \cos\phi_e - \sin^2\phi_e\right) - k_\theta \sin^2\phi_e \\ &\quad \le -\frac{k_r \sin r_e'}{r_e} \cos\phi_e \left(r_e^2 \cos\Phi_m - \sin^2\Phi_m\right) - k_\theta \sin^2\phi_e\end{aligned}$$

which is not a function of θ_e. Since r_e, $\sin r_e'$ and $\cos\phi_e$ are positive, and r_e is greater than $(\sin\Phi_m \tan\Phi_m)^{1/2}$ in Region_2 in the case of $\cos\Phi_m \le ((1+(R_e^2/4))^{1/2} - (R_e/2))$, then $\dot{V}(r_e, \phi_e) < 0$. Thus, r_e and ϕ_e will converge toward zero so that the robot in Region_2 moves to Region_1 or Region_3.

Case 3: In the case of Region_1, $-\Phi_m \le \phi_e \le \Phi_m$ and $R_e \le r_e \le R_a$.

Since $d = 1$ and $\epsilon = 0.5 - \alpha \mathrm{Sgn}(\sin\theta_e)\mathrm{Sgn}(\sin\theta_e + \sin\phi_e)$, the control commands are functions of r_e, ϕ_e, and θ_e. The derivative of the Lyapunov function is

$$\begin{aligned}\dot{V}(r_e, \phi_e, \theta_e) = &-\frac{k_r \sin r_e'}{r_e} \cos\phi_e \left(r_e^2 \cos\phi_e - \sin^2\phi_e\right) \\ &- k_\theta \epsilon (\sin\phi_e + \sin\theta_e)(\sin\phi_e + \beta \sin\theta_e)\end{aligned}$$

where $\beta = (1 - \epsilon)/\epsilon$. The first term in the above equation is negative, as was shown in **Case 2**.

In order to prove negativeness, the second term on the right-hand side is classified into four cases.

1) If $\sin\theta_e \ge 0$ and $\sin\phi_e + \sin\theta_e \ge 0$, the second term is negative, since $\beta > 1$ and $\sin\phi_e + \beta\sin\theta_e > 0$.
2) If $\sin\theta_e < 0$ and $\sin\phi_e + \sin\theta_e \ge 0$, it is negative, since $\beta < 1$ and $\sin\phi_e + \beta\sin\theta_e > 0$.
3) If $\sin\theta_e \ge 0$ and $\sin\phi_e + \sin\theta_e < 0$, it is negative, since $\beta < 1$ and $\sin\phi_e + \beta\sin\theta_e < 0$.
4) If $\sin\theta_e < 0$ and $\sin\phi_e + \sin\theta_e < 0$, it is negative, since $\beta > 1$ and $\sin\phi_e + \beta\sin\theta_e < 0$.

Therefore, since the second term is negative in all cases except at planes $(0, \phi_e, -\phi_e)^T$ and $(0, \phi_e, \sin^{-1}((-1/\beta)\sin\phi_e))^T$, $\dot{V}$ is a negative semi-definite function in Region_1. This means that the error trajectories converge toward a subset of the planes $(0, \phi_e, -\phi_e)^T$ or $(0, \phi_e, \sin^{-1}((-1/\beta)\sin\phi_e))^T$. Since V is a *locally positive definite function* and $\dot{V}$ is a *negative semi-definite function*, this system is stable. As the planes $(0, \phi_e, -\phi_e)^T$ and $(0, \phi_e, \sin^{-1}((-1/\beta)\sin\phi_e))^T$ are invariant sets where $\dot{V} = 0$, all the system phase trajectories converge toward one of the points asymptotically, as per the local invariant set theorem [15]. To show that the unique convergent point is the origin $(0, 0, 0)^T$ among the points in the planes $(0, \phi_e, -\phi_e)^T$ and $(0, \phi_e, -\sin^{-1}((-1/\beta)\sin\phi_e))^T$, the proposed control laws are substituted into the configuration error (4)

$$\dot{r_e} = -d\cos\phi_e \left(k_r \sin r_e'\right) \cos\phi_e \tag{10}$$

$$\begin{aligned}\dot{\phi_e} = &\frac{d \sin\phi_e}{r_e} \cos\phi_e \left(k_r \sin r_e'\right) - k_\theta \epsilon \sin\phi_e \\ &- k_\theta (1 - \epsilon) \sin\theta_e\end{aligned} \tag{11}$$

$$\dot{\theta_e} = -k_\theta \sin\phi_e \epsilon - k_\theta \sin\theta_e (1 - \epsilon). \tag{12}$$

From (10)-(12), it is found that none of the points, except $(0, 0, 0)^T$, is an equilibrium point. Therefore, the errors converge to $(0, 0, 0)^T$ asymptotically, and the robot moves to Region_0.

Case 4: In the case of Region_0, $r_e \le R_e$.

If $\epsilon = 0$, the motion controller command is a function of r_e and θ_e. In the region, θ_e will converge to 0. However, the command cannot guarantee the convergence of r_e and ϕ_e.

As a result, a robot located at any of the four regions will move to Region_0 with a predetermined orientation according to the proposed control law. ■

REFERENCES

[1] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, R. W. Brockett, R. S. Millman, and H. J. Sussman, Eds. Boston, MA: Birkhauser, 1983, pp. 181–191.

[2] C. C. de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, *Nonlinear Control Design for Mobile Robots, Recent Trends in Mobile Robots; World Scientific Series in Robotics and Automated Systems*, Y. F. Zheng, Ed, Singapore: World Scientific, 1993, vol. 11.

[3] E. P. Silva, F. L. Pereira, and J. B. Sousa, *On the Design of a Control Architecture for an Autonomous Mobile Robot, Advances in Intelligent Autonomous Systems (IAS).* Norwell, MA: Kluwer, 1998.

[4] B. D'Andrea-Novel, G. Bastin, and G. Campion, "Modeling and control of nonholonomic wheeled mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 1130–1135.

[5] Y. Kanayama and S. Y. Yuta, "Vehicle path specification by a sequence of straight lines," *IEEE Trans. Robot. Automat.*, vol. 4, pp. 265–276, Apr. 1988.

[6] J. Crowley, "Asynchronous control of orientation and displacement in a robotic vehicle," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1989, pp. 1277–1282.

[7] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," *Robot. Res.*, vol. 3, pp. 333–340, 1986.

[8] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1990, pp. 384–389.

[9] M. Aicardi, G. Casalino, A. Balestrino, and A. Bicchi, “Closed-loop smooth steering of unicycle-like vehicles,” in *Proc. IEEE Int. Conf. Decision and Control*, 1994, pp. 2455–2458.

[10] J.-M. Yang and J.-H. Kim, “Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots,” *IEEE Trans. Robot. Automat.*, vol. 15, pp. 578–587, June 1999.

[11] A. Astolfi, “Exponential stabilization of a wheeled mobile robot via discontinuous control,” *J. Dynam. Syst., Meas., Control*, vol. 121, pp. 121–126, Mar. 1999.

[12] R. T. M’Closkey and R. M. Murray, “Exponential stabilization of driftless nonlinear control systems using homogeneous feedback,” *IEEE Trans. Automat. Contr.*, vol. 42, pp. 614–628, May 1997.

[13] H.-S. Shim and J.-H. Kim, “Robust adaptive control for nonholonomic wheeled mobile robot,” in *Proc. IEEE Int. Conf. Industrial Technology*, Guangzhou, China, 1994, pp. 245–248.

[14] Y. Zhao and M. Reyhanoglu, “Nonlinear control of wheeled mobile robots,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 1967–1973.

[15] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

Path-Tracking of a Tractor-Trailer Vehicle Along Rectilinear and Circular Paths: A Lyapunov-Based Approach

A. Astolfi, P. Bolzern, and A. Locatelli

***Abstract*—The problem of asymptotic stabilization for straight and circular forward/backward motions of a tractor-trailer system is addressed using Lyapunov techniques. Smooth, bounded, nonlinear control laws achieving asymptotic stability along the desired path are designed, and explicit bounds on the region of attraction are provided. The problem of asymptotic controllability with bounded control is also addressed.**

***Index Terms*—Articulated vehicles, autonomous vehicles, Lyapunov design, mobile robots, nonlinear stabilization.**

I. Introduction

This paper addresses the problem of designing a controller for a tractor-trailer autonomous vehicle which has to follow a prescribed path. The only control variable is the steering angle of the tractor front wheels. The desired path either consists of a circle of a given radius or of a straight line to be followed at a specified speed, in either a forward or backward maneuver. Observe that restricting the attention to circular and rectilinear paths is not too severe a limitation, since any path can be suitably approximated by a sequence of circular and rectilinear arcs.

This problem is of great interest, both in applications and in purely theoretical contexts. On one side, its solution provides the potential for automatic guidance of a large class of industrial articulated vehicles, such as mining trucks, earth-removal and road-paving vehicles, buses for intercity travels, automated guided vehicles (AGVs), etc. (see, e.g., [1]–[6]). On the other side, the problem has constituted a challenging benchmark for testing the effectiveness of several advanced nonlinear control techniques (see, e.g., [7]–[20]). Most of the methods based on either Jacobian or input–output linearization [17], [20] have the drawback that convergence to the prescribed path is ensured only if the vehicle initial configuration is sufficiently close to the desired one, and the size of the region of attraction is difficult to evaluate. On the other hand, exact state-feedback linearization [11] yielding global convergence is effective only in the case of on-axle hitching.

More advanced nonlinear control techniques, such as those based on chained form [12], [13], [21], [22] or flatness [7], [8], [18] require, in general, a nonobvious selection of the guidepoint and the simultaneous use of two control variables, namely, the longitudinal velocity and the steering velocity. Moreover, they are generally not robust with respect to uncertainty in the system parameters.

This paper makes use of Lyapunov techniques as a tool for the design of a path-tracking controller for the tractor-trailer vehicle. The guidepoint is located in the middle of the tractor rear axle, and the approach of [9] is followed to decouple geometric path tracking from the velocity control. Therefore, independent design of the longitudinal velocity controller and the steering controller is possible. Herein, only the steering controller will be considered. The purpose of the paper is twofold. First, to cope in a unified framework with the general case of positive/null/negative off-axle distance. Second, to design control laws which allow for a precise characterization of the stability domain of the closed-loop system. Furthermore, all the designed controllers take possible limitations on the control action (saturation of the steering angle) explicitly into account.

The paper is organized as follows. Section II states the path-tracking control problem and introduces the model of the path-tracking offsets. The construction of stabilizing control laws is presented in Section III, while Section IV contains some simulation results. The paper ends with some concluding remarks in Section V.

II. Path-Tracking Offsets Model

The vehicle (see Fig. 1) consists of a wheeled tractor with two rear-drive wheels and a front steering wheel, towing a trailer, possibly with off-axle hitching ($c \neq 0$). The off-axle length c has to be regarded as a variable with the sign being negative when the joint is in front of the wheel axle, and positive otherwise. The longitudinal speed v_1 and the steering angle δ of the tractor are the control variables to be (independently) manipulated so that the guidepoint P_1 follows a desired path with an assigned velocity. We will be concerned with two particular yet significant cases, a rectilinear path and a circular path of radius R_1. For simplicity, it is assumed that the path must be followed at constant speed, but an appropriate time scaling can be used to deal with the (more general) case of variable, yet sign-definite, speed.

Let $l_{\mathrm{os}}, \vartheta_{\mathrm{os}}$ denote the tractor lateral offset and its orientation offset, respectively (see Fig. 1). They are measured with reference to the projection of the point P_1 of the tractor onto the path. Moreover, let $\varphi_{\mathrm{os}} = \varphi - \varphi_p$ be the difference between the current angle φ between tractor and trailer and its steady-state value φ_p along the prescribed path. Path tracking can be viewed as the task of driving these offsets asymptotically to zero.

Manuscript received June 10, 2002; revised January 29, 2003. This paper was recommended for publication by Associate Editor H. Arai and Editor A. De Luca upon evaluation of the reviewers’ comments. This work was supported in part by Consiglio Nazionale delle Ricerche (CNR), Italy, the Ministerio dell’Istruzione, dell’Università e della Ricerca Scientifica (MIUR) project “Identification and adaptive control of industrial systems,” and the NAC02 (Nonlinear and Adaptive Control) TMR European Programme. This paper was presented in part at the European Control Conference, Porto, Portugal, September 2001.

The authors are with the Politecnico di Milano, Dipartimento di Elettronica e Informazione, 20133 Milano, Italy (e-mail: astolfi@elet.polimi.it; bolzern@elet.polimi.it; locatelli@elet.polimi.it). A. Astolfi is also with the Department of Electrical and Electronic Engineering, Imperial College, London, U.K. (e-mail: a.astolfi@ic.ac.uk).

Digital Object Identifier 10.1109/TRA.2003.820928