Proceedings of the 8th IEEE Mediterranean Conference
on Control and Automation (MED 2000)
Rio, Patras, GREECE 17-19 July, 2000

EXPERIMENTAL COMPARISON OF CONTROL LAWS FOR UNICYCLE-TYPE MOBILE ROBOTS

BYUNGMOON KIM, PANAGIOTIS TSIOTRAS †

† Georgia Institute of Technology, School of Aerospace Engineering
Atlanta, GA, 30332-0150, USA, p.tsiotras@ae.gatech.edu

Abstract. Mobile robots offer a typical example of a system with a nonholonomic constraint. Many control laws have been developed for stabilizing these systems. One of the main issues with these controllers is that they are usually based on kinematic relations only and do not include the dynamics. Moreover, additional factors like quantization, noise and delay may be present that make stabilization more difficult. Comparing the characteristics and the performance of these controllers using an experimental testbed is therefore of great interest. In this paper, we use a Khepera robot to perform these experiments and compare several controller proposed in the literature.

Key Words. Nonholonomic systems, nonlinear control, mobile robots.

1. INTRODUCTION

Several examples which involve nonholonomic constraints can be found in real-world applications, like mobile robots, bicycles, cars or underactuated axi-symmetric spacecraft. Several control laws have been proposed by many researchers for stabilizing such systems. One group of researchers have used time-invariant, non-smooth controllers; see [8], [2], [1] and [9]. Another approach, followed in [5], [4], [7], [3], is to use time-varying controllers. Experimental validation of time-varying controllers can be found in [4]. However, a comparative study of controllers for nonholonomic systems, in particular, between time-varying and time-invariant controllers, has not been done so far. This is clearly of great interest. In this paper, we implement these controllers on a unicycle-type robot called Khepera. This robot has two DC-motor-powered wheels and introduces many realistic difficulties such as different motor dynamics in the two wheels, time delay, quantization, sensor noise and saturation. We apply various control laws and test their performance with respect to these issues. Some ways to improve the robot performance is also discussed. The proofs of stability are omitted either because they are available in the literature or because of space limitations. In the latter case, the proofs are available from the authors upon request.

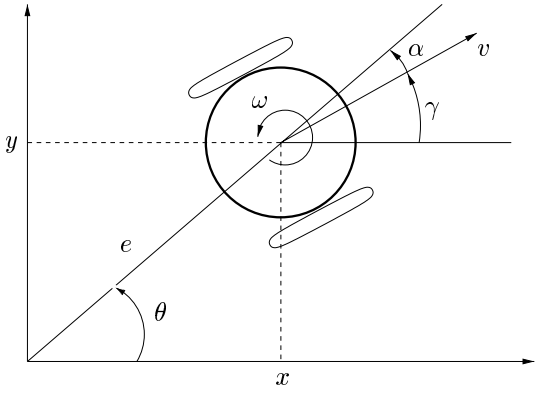

Fig. 1. Definition of configuration variables.

2. KINEMATIC EQUATIONS

Consider a unicycle-type robot with two wheels, as shown in Fig. 1. The kinematic equations of the mobile robot are

$$\dot{x} = v\cos\gamma, \quad \dot{y} = v\sin\gamma, \quad \dot{\gamma} = \omega \tag{1}$$

The kinematic model of the mobile robot has two control inputs: velocity v and angular velocity ω (see Fig. 1). The velocities of the left and right wheels are given by $v_1 = v - R\omega$ and $v_2 = v + R\omega$, respectively, where R is the distance between the two wheels. Equation (1) can be transformed to the normal form of a nonholonomic system in chained (or power) form by a state and input transformation. There are two cases, depending on the input transformation used. Using the state transformation

$$x_1 = x\cos\gamma + y\sin\gamma\,, \quad x_2 = \gamma \tag{2}$$
$$x_3 = x\sin\gamma - y\cos\gamma \tag{3}$$

System I is given by Eq. (4) and System II is given by Eq. (5).

$$\text{(I)} \quad \begin{cases} \dot{x}_1 = u_1\,, \quad \dot{x}_2 = u_2\,, \quad \dot{x}_3 = x_1 u_2 \\ v = u_1 + x_3 u_2\,, \quad \omega = u_2 \end{cases} \tag{4}$$

$$\text{(II)} \quad \begin{cases} \dot{x}_1 = u_1 - x_3 u_2\,, \quad \dot{x}_2 = u_2\,, \quad \dot{x}_3 = x_1 u_2 \\ v = u_1\,, \quad \omega = u_2 \end{cases} \tag{5}$$

The two systems differ by the term $x_3\omega$ in the $\dot{x}_1$ equation. Controller performance depends on the system used. System I is the transformation on which many controllers have been developed, while System II often gives better results in practice. As was shown in [4], the extra term $x_3\omega$ does not destroy stability.

3. TIME-INVARIANT CONTROLLERS

In this section, we will discuss four choices of time-invariant controllers. Three of them use ideas from invariant manifold theory [8], [2] and [9]. The other controller is designed using Lyapunov techniques [1]. In this controller, no state transformation is used. The benefit of this will be discussed in the sequel.

3.1 Controller 1

This controller, given by Eq. (6) and (7), is taken from [8]. It is given by

$$u_i = -kx_i + \beta_i(x,s) \qquad i = 1,2 \tag{6}$$

where β_1 and β_2 are designed such as

$$\beta_1(x,s) = \mu\frac{s(x)x_2}{x_1^2 + x_2^2}\,, \quad \beta_2(x,s) = -\mu\frac{s(x)x_1}{x_1^2 + x_2^2} \tag{7}$$

The control input given by Eqs. (6) and (7) globally asymptotically stabilize System I for all initial conditions such that $x_1(0) \neq 0$ and $x_2(0) \neq 0$, if $\mu > 0$ and $k > 0$. Moreover, the control input u_1 and u_2 are bounded along the trajectories of the closed-loop system, if $\mu > 2k > 0$. For the complete proof, refer to [2].

The controller given by Eqs. (6)-(7), is not defined on x_3 axis where $x_1 = x_2 = 0$. We call this the *singular case*. Moreover, $\beta_{1,2}$ become large as the initial condition approaches to the *singular case*, resulting to motor saturation. We expect that the closeness of the initial condition to the *singular case* is a measure of the difficulty the controller has to stabilize the system using small control inputs. To measure this difficulty, we define η by

$$\eta \equiv \sqrt{\beta_1^2 + \beta_2^2} = \frac{|s|}{\sqrt{x_1^2 + x_2^2}} \tag{8}$$

Using η, we can define a so-called *difficult region* by $\eta \geq \eta_0$. This is essentially a non-trivial, closed neighborhood of the singular manifold. The value of η_0 is chosen by experimentation. For the *difficult region* where η is large, we need to apply some control law to escape from this region. Since the robot can escape from the *singular case* by almost any control law, the same is true for the *difficult region*, provided that this region is small enough. We propose the simple control law for this region $u_1 = k_s\mathrm{sgn}(s)$ and $u_2 = 0$. The gain k_s is chosen by experimentation or simulation. It can be shown that once the trajectory has escaped from the *difficult region*, it will not enter this region again. One may verify this from the fact that $\dot{\eta} = -(\mu - 2k)/2\,\eta$ for the closed-loop system under the control law in (6) and (7).

Now, we turn our attention to System II. Local stability with the control law in (6)-(7) can be proved using results from the theory of homogeneous systems [6].

Proposition 1. The control laws given by Eq. (6) and (7) locally asymptotically stabilize the System II, if $k > 0$ and $\mu > 0$.

3.2 Controller 2

The second controller also uses invariant manifold ideas but it provides an explicit bound on the control input. This control law was originally developed for the stabilization problem of an underactuated axisymmetric spacecraft [9]. It is modified here for the case of a mobile robot. In this controller, the control input is bounded by some finite value regardless of the initial conditions. The control law is given by

$$u_i = -k\frac{x_i}{\sqrt{\nu^2 + 1}} + \mu\,\mathrm{sat_i}(s,\nu), \quad i = 1,2 \tag{9}$$

where $\nu = \sqrt{x_1^2 + x_2^2}$. The saturation functions sat_1 and sat_2 are defined as

$$\mathrm{sat}_{1,2}(s,\nu) = \begin{cases} \mathrm{sat}\left(\frac{s}{\nu}\right)\frac{x_{2,1}}{\nu}, & \text{if } \nu \geq \epsilon \\ \mathrm{sgn}(s), & \text{if } \nu < \epsilon \end{cases} \tag{10}$$

where $\epsilon \ll 1$ is introduced to avoid chattering.

Proposition 2. The control law given by Eqs. (9) and (10) with

$$\mu > 2k > 0, \text{ if } |s|/\nu < 1 \quad (11)$$
$$\mu > -2k > 0, \text{ if } |s|/\nu \geq 1 \quad (12)$$

globally asymptotically stabilizes System I. Moreover, the control input is bounded by $|u_{1,2}| \leq |k| + |\mu|$.

In System I, this controller does not produce bounded v, but for System II, it provides bounded v and ω. The following Proposition shows that this controller also works for System II.

Proposition 3. For $k > 0$ and $\mu > 0$, the control law in Eq. (9) and (10) globally asymptotically stabilizes System II.

3.3 Controller 3

The third controller is a switching controller, also based on invariant manifold ideas. This controller was proposed by Khennouf and Canudas de Wit in [2]. This two-stage switching structure controller is given below

$$\begin{aligned} u_1 &= +\sigma \operatorname{sgn}(s)\, x_2 \\ u_2 &= -\sigma \operatorname{sgn}(s)\, x_1 \end{aligned} \quad \text{for } |s| \geq \epsilon \quad (13)$$

and

$$u_1 = -kx_1\ ,\ \ u_2 = -kx_2 \quad \text{for } |s| < \epsilon \quad (14)$$

where $k > 0$ and $\sigma > 0$. The reader may refer to [2] for the proof of the stability. This controller will not work if $x_1 = x_2 = 0$. This is the same *singular case* as in Controller 1. Note that if $x_1^2 + x_2^2$ is small, the control input will be small, resulting to a deadzone problem (assuming that $x_3 \neq 0$). Like Controller 1, we can expand the *singular manifold* to a closed neighborhood and use a simple control law to escape initial conditions in this region. For System II, it was observed that this controller exhibits significant chattering and very slow convergence rates.

3.4 Controller 4

Another time-invariant controller was proposed in [1]. This controller is given directly in terms of the physical state-space representation of the kinematics. No transformation to chained or power normal form is necessary. This control law is given by

$$v = -k_1\, e\, \cos\alpha \quad (15)$$
$$\omega = k_2\alpha + k_1 \frac{\cos\alpha \sin\alpha}{\alpha}(\alpha + k_3\theta) \quad (16)$$

where, $k_1 > 0$, $k_2 > 0$, $k_3 > 0$, e is the distance from the robot to the origin, and x and y are the coordinates of the position of the robot, as shown in Fig. 1. The stability of this system is proved in [1].

4. TIME-VARYING CONTROLLERS

Several time-varying controllers have been developed for the nonholonomic mobile robot but their performance is still under investigation. Some experimental results have been reported in [4]. In this work we choose four time-varying controllers and evaluate their performances. The reader may refer to the relevant references for the complete proofs of these controllers.

4.1 Controller 5

A time-varying controller for the chained normal form in System I using sinusoids is proposed in [7]. The controller is given by

$$u_1 = -x_1 - x_3^2(\sin(t) - \cos(t)) \quad (17)$$
$$u_2 = -x_2 - c_1 x_3 \cos(t) \quad (18)$$

For the implementation of this controller, System I was used.

4.2 Controller 6

Another time-varying controller was proposed in [3,4]. This controller is given by

$$u_1 = -c_{11}x_1 + c_{12}\frac{x_3}{\rho(x)}\cos(\Omega t) \quad (19)$$

$$u_2 = -c_{21}x_2 + c_{22}\frac{x_3^2}{\rho(x)^3}\sin(\Omega t) \quad (20)$$

where $c_{ij} > 0$ and $\rho(x) \equiv (x_1^4 + x_2^4 + x_3^2)^{\frac{1}{4}}$. Use of System I often resulted in unstable response. Thus, System II was used to implement this controller, which improved the convergence properties.

4.3 Controller 7

The second time-varying controller, experimentally evaluated in [4], is given by

$$u_1 = -c_{11}x_1 + c_{12}x_3\cos(\Omega t) \quad (21)$$
$$u_2 = -c_{21}x_2 + c_{22}x_3^2\sin(\Omega t) \quad (22)$$

For the implementation of this controller, System II is used. The reader may refer to Ref. [4] for further discussion on this controller.

4.4 Controller 8

One of the first time-varying controllers for the chained normal form in System I, was the one proposed by Pomet et al. in [5]. This controller is given by

$$u_1 = x_3\sin(t) - (x_1 + x_3\cos(t)) \quad (23)$$
$$u_2 = -(x_1 + x_3\cos(t))x_1\cos(t) - (x_1x_3 + x_2) \quad (24)$$

For the implementation of this controller, System I was used.

5. CONTROLLER IMPLEMENTATION

The implementation of the controllers discussed previously was done on a Khepera mobile robot. Khepera is a small-size robot developed for educational and research purposes by K-Team (see the

URL `http://www.k-team.com` for more information). It has several proximity sensors (not used in this work) and can work in semi-autonomous (server-client) or completely autonomous mode. In the server-client mode the robot is controlled through an RS-232 serial port by a host computer. A specially written C++ application running under Windows NT was developed to implement the previous algorithms and control the robot.

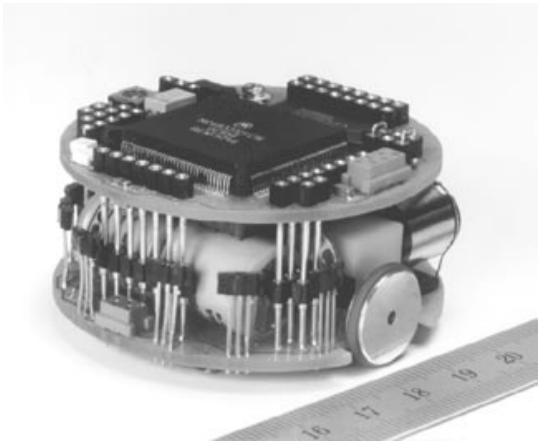

Fig. 2. The Khepera robot.

5.1 The Khepera Robot

The Khepera mobile robot uses two DC motor driven wheels. The DC motors are connected to the wheels through a 25:1 reduction gear box. Two incremental encoders are placed on the motor axes. The resolution of the encoder is 24 pulses per revolution of motor axis. This corresponds to $24 \times 25 = 600$ pulses per revolution of the wheels or 12 pulses per millimeter of wheel displacement. The algorithm to estimate the velocity from the encoder output is implemented on the robot. For the DC motor speed control, a native PID controller is also implemented on the Khepera robot. All one needs to do in order to control Khepera, is to read position signals and issue velocity commands via the RS-232 serial port.

5.2 Implementation of Controllers on a WinNT Environment

C/C++ is used to implement the previous control algorithms with a nice-looking, multi-tabbed dialog box interface, shown in Fig. 3. From the `Realtime` tab, one can click the target position and orientation. The software automatically sets up a stabilizing problem by transforming the target position/orientation to origin and current position/orientation to the initial conditions. For the discrete implementation of the continuous controller, a 32bit multimedia timer service in NT is used and all other applications are closed to minimize the timer latency. The software provides a combo-box interface to select the sampling frequency of the controller in the `Configuration` tab. The maximum sampling rate can, theoretically, be slightly over 100Hz because of the speed limitation of the RS-232 serial communication (maximum is 4.8kbytes/s in the Khepera robot). For all experiments in this paper, we choose 50Hz for the sampling frequency. The software also features a `Sensors` display tab, and a `Console` tab. The motor can be tested/configured via the `Performance` tab. To easily change the gains of the controller, all gains and other parameters are stored in a `.ini` file. The software comes also with an `ini` editor so that the user can change the settings online. To record the history of the control input and robot response without recording time limitations, a double-buffered data storage algorithm was developed. The robot can also be visualized by an independent OpenGL Window that supports 6DOF camera navigation using the keyboard.

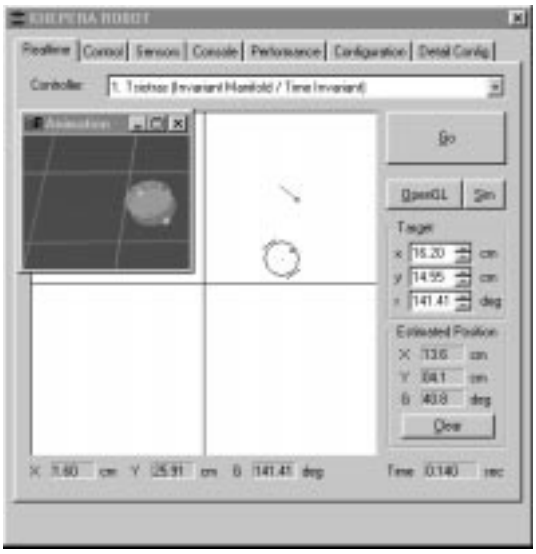

Fig. 3. Robot control program interface.

5.3 Estimation of Position

Since there is no information about the absolute position and orientation of the robot, *dead-reckoning* is used to find the current position and orientation from a known position and orientation. This method is simple, but its main drawback is that the error is accumulated over long periods of time. For short time maneuvers, like the ones in this paper however, it is quite adequate. To calibrate the robot we run several open-loop maneuvers on a one-millimeter resolution grid paper. The position estimation error was verified to be less than 10% in any given mission.

5.4 Quantization in the Velocity Output

The velocity command of the Khepera robot is quantized by 8 mm/sec. At the origin, quantization

is manifested as a dead-zone problem. If we apply a control law, the small velocity command will be ignored by the robot and we will get large steady state errors. For example, the steady state error in γ was about $15 \sim 60$ deg with Controllers 1,2 and 4. One simple but effective approach is to use an inverted dead-zone to handle this problem. As shown in Fig. 4, all magnitudes of velocity commands are increased by the amount of quantization. This will make the error small. After achieving a certain value of the error, the error will not decrease any more but the system will remain in a limit cycle. The inverted dead-zone is implemented in software.

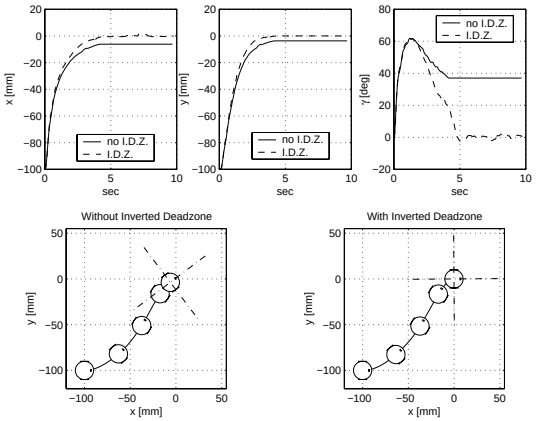

Fig. 4. Effect of inverted deadzone on the steady state error.

5.5 Scaling

Before implementing the controllers to the real robot, we need to choose the units of several variables, or more generally, to scale the states. By choosing the scaling value, we can adjust the magnitude of the state values. For example, System I with Controller 1 has large overshoot around the origin. As a result, if the initial condition is $x_{10} = 0.1$ m, $x_{30} = 0.05$ m and $x_{20} = 0$ deg, the system will exhibit large oscillations during transients. To improve the performance, we can scale the state to a large enough value so that the oscillation can be significantly reduced. Figure 5 shows the improvement when the states are scaled to $x_{10} = 100$ mm, $x_{30} = 50$ mm and $x_{20} = 0$ deg. For each controller, state scaling was chosen to give the best transient response.

5.6 Effect of Sensor Noise and Quantization on the Steady State Error

Sensor noise and quantization limits control accuracy. Typically, sensor noise and quantization (deadzone around the origin) will result in a limit cycle. In both experiments and simulations, Controller 4 showed large magnitude of the limit cycle. This is caused by the control law that determines the steering angle regardless of the distance from

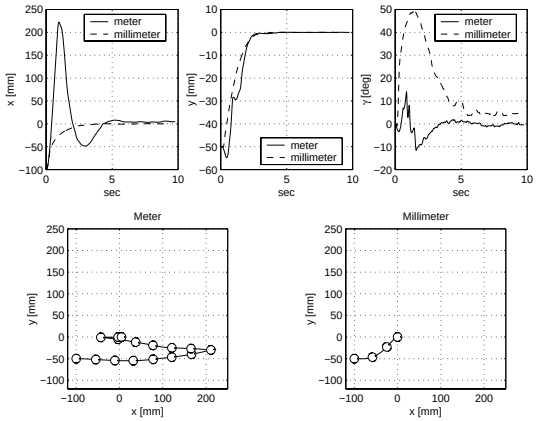

Fig. 5. Comparison of responses with different units (scaling).

the origin (notice that the ω command in (15) is independent of e). Even after converging close enough to the origin, Controller 4 still produces a large heading angle. Figure 6 shows that Controller 4 has a large noise response in γ compared to the other controllers.

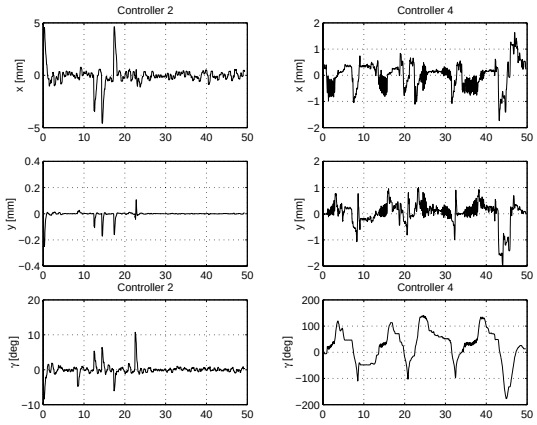

Fig. 6. Noise effects on the steady state error.

6. EXPERIMENTAL RESULTS

6.1 Mission Design

To compare the different controllers, we designed four missions: easy, normal, singular and long distance. We defined the difficulty of each mission by the ratio between forward and sideways motions. The initial value of γ is chosen to be zero. A long distance mission was devised to demonstrate the motor saturation due to possible large control commands. During the long distance mission the advantage of Controller 2 which has bounded input was evident. The initial conditions for these missions are shown in Table 1. Starting from this position, the robot is commanded to move to the origin.

Table 1. Mission specifications.

Mission	x (mm)	y (mm)	γ (deg)
Easy	-100	-25	0
Normal	-100	-100	0
Singular	0	-100	0
Long Distance	-500	-500	0

6.2 Experimental Results

The summary of the experiments are shown in Table 2. In this table, 'E', 'N', 'S' and 'L' stand for easy, normal, singular and long distance missions, respectively. 'G' stands for 'Good', which means that the convergence is fast enough, i.e., within 10 seconds. 'S' stands for 'Slow', which means that it took more than 20 seconds to converge. 'O' stands for oscillatory, which means that the trajectory did not converge nor diverge but oscillated around the origin. The detailed plots of the results are available from the authors upon request.

Table 2. Experimental results.

Ctr.	Sys.	E	N	S	L	Note
1	I	G	O	O	O	Oscillatory
1	II	G	G	G	G	Good
2	I	G	O	O	O	Oscillatory
2	II	G	G	G	G	Good
3	I	G	G	G	O	Chattering
4	-	G	G	G	G	Noise Sensitive
5	I	G	S	S	O	Slow
6	II	S	S	S	S	Slow
7	II	S	S	S	S	Slow
8	I	S	S	S	O	Slow

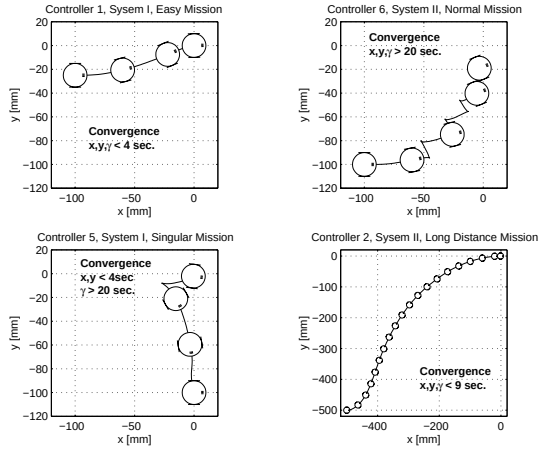

Fig. 7. Trajectories of some experiments

7. CONCLUSIONS

In this paper, we have experimentally tested several controllers for a unicycle-type mobile robot. Several controllers resulted in oscillatory or even unstable behavior. To improve the controller performance, we implemented different state and control transformations, scalings and inverted deadzone. By applying these techniques, the performance of Controllers 1 and 2 improved enormously. These controllers showed the best overall performance. Controller 3 turned out to be impractical because of excessive chattering. Controller 4 showed good performance except for the noise sensitivity around the origin. Controller 6 failed to achieve stability for the actual robot when using System I. Nonetheless, the use of System II achieved better convergence properties. The time-varying controllers were very slow and oscillatory. Further experiments using actual mobile robots are needed to verify these initial assessments.

8. REFERENCES

[1] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, Vol. 2, pp. 27–35, 1995.

[2] H. Khennouf and C. Canudas de Wit, "On the construction of stabilizing discontinuous controllers for nonholonomic systems," in *IFAC Nonlilear Control Systems Design Symposium*, pp. 747–752, 1995. Tahoe City, CA.

[3] R. T. M'Closkey and R. M. Murray, "Exponential convergence of nonholonomic systems: some analysis tools," in *Proceedings of the 31st Conference on Decision and Control*, pp. 943–948, 1993. San Antonio, Texas.

[4] R. T. M'Closkey and R. M. Murray, "Experiments in exponential stabilization of a mobile robot towing a trailer," in *Proceedings of the American Control Conference*, pp. 988–993, 1994. Baltimore, Maryland.

[5] J. B. Pomet, "Explicit design of time-varying stabilizing control laws for a class of controllable systems without drift," *Systems and Control Letters*, Vol. 18, pp. 147–158, 1992.

[6] L. Rosier, "Homogeneous Lyapunov function for homogeneous continuous vector field," *Systems and Control Letters*, Vol. 19, pp. 467–473, 1992.

[7] A. R. Teel, R. M. Murray, and G. Walsh, "Nonholonomic control systems : From steering to stabilization with sinusoids," in *Proceedings of the 31st Conference on Decision and Control*, pp. 1603–1609, 1992. Tuckson, Arizona.

[8] P. Tsiotras and J. Luo, "Invariant manifold techniques for control of underactuated mechanical systems," in *Proceedings of the Workshop Modeling and Control of Mechanical Systems, Ed: A. Astolfi, P. J. N. Limebeer, C. Melchiorri, A. Tornambe, R. R. Vincer*, pp. 277–292, 1997. London, World Scientific Publishing Co.

[9] P. Tsiotras and J. Luo, "Stabilization and tracking of underactuated axisymmetric spacecraft with bounded control," in *IFAC Symposium on Nonlinear Control Systems Design (NOLCOS'98)*, 1998. Enschede, The Netherlands.