

## Experiments in Visual Feedback Control of a Wheeled Mobile Robot \*

Alessandro De Luca Giuseppe Oriolo Luca Paone Paolo Robuffo Giordano

Dipartimento di Informatica e Sistemistica  
Università degli Studi di Roma "La Sapienza"  
Via Eudossiana 18, 00184 Roma, Italy  
{deluca,oriolo}@dis.uniroma1.it

### Abstract

*An experimental study is presented on vision-based feedback control methods for the nonholonomic wheeled mobile robot SuperMARIO. The robot posture is measured via a camera fixed on the ceiling of an indoor environment. To this end, a simple localization algorithm has been developed. Performance on trajectory following and parking tasks is compared under different controllers and using either odometric or visual feedback. The improvement with the latter is obtained at the expense of a limited increase in sampling time.*

### 1 Introduction

Many feedback controllers have been proposed in the last decade for nonholonomic Wheeled Mobile Robots (WMRs) [1], addressing the basic motion tasks in a free indoor environment, i.e., following a desired cartesian trajectory or parking at a given configuration.

Control methods for stable trajectory tracking include linear control based on tangent linearization, nonlinear control based on Lyapunov techniques [2, 3], and dynamic feedback linearization [4, 5]. The search for stabilizing controllers in parking tasks has led to novel solutions, based on time-varying [6] and/or discontinuous nonlinear feedback [7], which overcome the well-known obstruction to the existence of smooth stabilizing controllers for first-order nonholonomic systems. Other control designs include the use of a polar coordinate transformation [8] or, again, of dynamic feedback linearization [9]. A detailed comparison of the actual performance of these techniques has been made in simulated environments, both for a unicycle-type vehicle [10] and a car-like vehicle [11], and through real experiments on a two-wheel differentially driven mobile robot [12].

Most of the performed experimental works assume the availability of the full current state of a WMR. The robot state, e.g., the position  $(x, y)$  and orientation  $\theta$

of a unicycle robot, is often reconstructed on the basis of proprioceptive sensors, e.g., incremental encoders mounted on the motor axes (dead reckoning). However, these odometric estimates are subject to random and systematic errors, due to wheel slippage during maneuvers and to non-idealities of mechanical parts (e.g., different wheel radius), as well as to possible numerical drift over time (integration process).

Exteroceptive sensing (ultrasonic, laser, vision) may provide instead an absolute measure of the robot state (localization). In particular, the information gathered by a vision system—the richest among exteroceptive sensors—about the environment makes it possible to detect natural landmarks, navigate among unknown obstacles, and achieve a reactive robot behavior. Vision-based sensing has also some drawbacks, such as the need to recognize and extract a number of characteristic features from the image, an increased computational burden, and a critical dependence on lightning conditions of the environment.

There are basically two ways of using exteroceptive information within a motion control loop: *i*) map measures back to the robot state space and then use state-based control; *ii*) close the feedback directly in the sensor space, thus realizing an output feedback strategy. In vision-based control, the two approaches are often referred to as (position-based) visual feedback and (feature-based) visual servoing, respectively. The first requires accurate calibration of the external sensors, but it allows a common ground for the fusion of heterogeneous sensor measures. The second relaxes the need for precise mechanics and is advantageous in less structured environments, but requires to deal with kinematic transformations and a redefinition of conventional, position-based control laws.

In this paper, we present experimental results on the use of visual information within a real-time motion control loop for a nonholonomic WMR. Previous work in this area includes, e.g., [13] for visual feedback and [14, 15, 16] for visual servoing. In particular, using the mobile robot SuperMARIO as a testbed, we have

\*Work supported by MURST within the *MISTRAL* project.

compared visual feedback control laws for trajectory following and parking tasks, reconstructing the robot state from images obtained by a camera fixed on the ceiling of an indoor environment. A discussion of the results achievable using odometric versus visual feedback will be given. We shall also illustrate how system constraints given by a common PC-based architecture affect the control design and the robot performance.

## 2 Experimental setup

The wheeled mobile robot is our prototype SuperMARIO, a two-wheel differentially-driven vehicle. The driving wheels have radius  $r = 9.93$  cm and are at a distance of  $d = 29$  cm along their axle, while a small passive off-centered wheel (castor) is placed near the vehicle front. The weight of the robot (including four 12 V batteries) is about 20 kg, with center of mass located slightly in front of the main axle so as to limit disturbances caused by the castor.

Each wheel is driven by a DC servomotor, supplied at 24 V and equipped with an incremental encoder counting 200 pulses/turn (multiplied by four through on-board electronics). Two 8-bit ST6265 microcontrollers implement a digital velocity PID on the left and right wheel, with a low-level cycle time of 5 ms, realizing the wheel angular velocity commands  $\omega_L$  and  $\omega_R$  generated by the high-level robot controller. SuperMARIO communicates via radiomodem with a 300 Mhz PC Pentium II, where a library of C++ control algorithms is installed.

### 2.1 Kinematic model

The kinematic model of the nonholonomic WMR SuperMARIO is equivalent to that of a unicycle:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \quad (1)$$

where  $(x, y)$  is the cartesian position of the wheel axle midpoint,  $\theta$  is the vehicle orientation w.r.t. an arbitrary  $x$ -axis, while  $v$  and  $\omega$  are, respectively, the linear and angular robot velocity. The actual input commands  $(\omega_L, \omega_R)$  are one-to-one related to  $(v, \omega)$ , by means of

$$v = r \frac{\omega_R + \omega_L}{2}, \quad \omega = r \frac{\omega_R - \omega_L}{d}. \quad (2)$$

### 2.2 Odometric feedback

One possible reconstruction of the current robot state is based on incremental encoder data (odometry). Let



Figure 1: A camera-image of SuperMARIO

$\Delta\phi_R$  and  $\Delta\phi_L$  be the angular wheel displacement measured by the encoders during a sampling time  $T_c$ . From eq. (2), the robot linear and angular displacements are

$$\Delta s = \frac{r}{2} (\Delta\phi_R + \Delta\phi_L), \quad \Delta\theta = \frac{r}{d} (\Delta\phi_R - \Delta\phi_L).$$

The posture estimated at time  $t_k = kT_c$  is then

$$\hat{q}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{bmatrix} = \hat{q}_{k-1} + \begin{bmatrix} \cos \bar{\theta}_k & 0 \\ \sin \bar{\theta}_k & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta s \\ \Delta\theta \end{bmatrix}, \quad (3)$$

where

$$\bar{\theta}_k = \hat{\theta}_{k-1} + \frac{\Delta\theta}{2}.$$

The localization of SuperMARIO using the odometric prediction (3) is quite accurate whenever wheel slippage does not occur and a small number of backup maneuvers is executed. Controllers based on odometric state reconstruction run with a sampling time  $T_c = 50$  ms (including reference motion generation).

### 2.3 Visual feedback

To provide visual feedback of the robot state, we have used a digital 1/2" camera with  $768 \times 576$  pixels, fixed on the laboratory ceiling. The camera has been calibrated using Tsai's algorithm [17]. The workspace (i.e., the vision area) dimensions are  $2.90 \times 2.10$  m, set by the height of the camera from the floor and by its focal distance. As a result, 1 pixel  $\approx 3.7$  mm. The camera output is a RGB or CCIR signal sent to a Matrox Meteor frame-grabber on the PC, with a 25 Hz frame rate in CCIR mode.

For reconstructing the robot posture by means of the camera, we have mounted on SuperMARIO a black

surface with three leds, located at the vertices of an isosceles triangle pointing in the forward direction (see Fig. 1). The upper vertex position is taken as the  $(x, y)$  reference point of the robot.

In order to localize the triangle in the image, we proceed as follows. A binary image is created first from a 256-level grayscale image, by using a threshold of 240. Using a *dilation* operator (see [18]), a more significant image is obtained from which a list of light *blobs* (in the range of 30–120 pixels each) is extracted. Unfortunately, reflections from the floor and the robot chassis (especially the wheels) are also marked as light blobs. A preliminary step to discard false reflections is to put upper and lower bounds to the blob area<sup>1</sup>. Next, an appropriate algorithm based on relative distances deletes from the list all the blobs that cannot be candidate vertices (within some tolerance), and builds with the remaining blobs all isosceles triangles with consistent side length. The localization routine fails if everything but a single triangle is returned (an extremely rare event in our experiments).

Once a single triangle is detected, let  $(x_u, y_u)$ ,  $(x_l, y_l)$ , and  $(x_r, y_r)$  be the coordinates of, respectively, the upper, lower-left, and lower-right vertex. The estimate of the robot reference point is simply  $\hat{x} = x_u$ ,  $\hat{y} = y_u$ . The center  $(x_c, y_c)$  of the triangle and its base midpoint  $(x_m, y_m)$  are then computed from the three vertices. The estimate  $\hat{\theta}$  (at time  $t_k$ ) of the robot orientation is finally computed as

$$\hat{\theta} = \frac{\theta_{uc} + \theta_{um} + \theta_{cm}}{3},$$

with  $\theta_{ij} = \text{ATAN2}\{y_i - y_j, x_i - x_j\}$ ,  $i, j \in \{u, c, m\}$ . We found out that this simple averaging strategy, which maximizes the use of measured data, is already effective in reducing the effects of image noise, that is particularly crucial in the reconstruction of  $\theta$ .

When using visual feedback, the control sampling time grows to  $T_c = 55$  ms, including frame acquisition, elaboration and robot-server communication. The increase in sampling time has been limited by making frame acquisition asynchronous from other control routines and by restricting the above triangle search to a  $300 \times 300$  window centered around the previous robot state estimate. A full window search would have led to a sampling time of 80 ms.

## 2.4 Motion constraints

The achievable precision of SuperMARIO is limited by the finite resolution of the digital low-level control

<sup>1</sup>This is done using the MIL Libraries that allow blob detection and fast analysis of basic blob features, such as center of mass, area, etc.

layer. In particular, the minimum linear displacement is  $\Delta s_{min} = 0.0039$  cm, while the minimum increment of linear velocity is 0.78 cm/s.

The motor dynamics imposes bounds on the maximum angular velocity and acceleration of the wheels. The robot may reach a maximum speed of about 2 m/s. However, in order to prevent as much as possible wheel slippage and robot rearing (especially at start), we have imposed conservative constraints on the linear velocity and acceleration of the wheels at their contact with ground:

$$|v_i| \leq v_{max} = 0.35 \text{ m/s}, \quad |a_i| \leq a_{max} = 0.83 \text{ m/s}^2,$$

with  $i \in L, R$ .

When reaching one of these saturation levels, it is necessary to perform a suitable command input scaling so as to preserve the curvature radius of the actual commanded motion. This is achieved by scaling in sequence first acceleration (if needed) and then velocity commands. For a given set of acceleration commands  $(a_R, a_L)$ , define

$$\sigma = \max \left\{ \frac{|a_R|}{a_{max}}, \frac{|a_L|}{a_{max}}, 1 \right\}.$$

We modify then the original commands as follows:

$$\begin{aligned} a_R &= \text{sign}(a_R) a_{max}, & a_L &= \frac{a_L}{\sigma} & \text{if } \sigma &= |a_R|/a_{max} \\ a_L &= \text{sign}(a_L) a_{max}, & a_R &= \frac{a_R}{\sigma} & \text{if } \sigma &= |a_L|/a_{max} \\ a_R, a_L &\text{ not modified} & & & \text{if } \sigma &= 1. \end{aligned}$$

From the obtained accelerations, left and right wheel velocities are computed at time  $t_k$  as  $v_{k+1} = a T_c + v_k$  and then scaled in a similar way using the velocity bound  $v_{max}$ .

## 3 Control laws

We briefly recall the five feedback control laws compared in our experiments, two for trajectory following and two for parking tasks.

### 3.1 Trajectory following

Let  $(x_d(t), y_d(t))$  be the reference cartesian trajectory on the workspace plane. The associated robot orientation is  $\theta_d(t) = \text{ATAN2}\{\dot{y}_d(t), \dot{x}_d(t)\}$  and the nominal feedforward commands are

$$\begin{aligned} v_d(t) &= \pm \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \\ \omega_d(t) &= \frac{\ddot{y}_d(t)\dot{x}_d(t) - \ddot{x}_d(t)\dot{y}_d(t)}{\dot{x}_d^2(t) + \dot{y}_d^2(t)}. \end{aligned}$$

**Nonlinear time-invariant control (NTI) [3]** The tracking controller is the outcome of a Lyapunov-based design:

$$\begin{aligned} v &= v_d \cos(\theta_d - \theta) + k_1 [(x_d - x) \cos \theta + (y_d - y) \sin \theta] \\ \omega &= \omega_d + k_2 v_d \frac{\sin(\theta_d - \theta)}{\theta_d - \theta} [(y_d - y) \cos \theta \\ &\quad - (x_d - x) \sin \theta] + k_3(\theta_d - \theta). \end{aligned}$$

An usual choice for the control gains is

$$k_1 = k_3 = 2\zeta\sqrt{\omega_d^2(t) + bv_d^2(t)}, \quad k_2 = b,$$

with damping coefficient  $\zeta \in (0, 1)$  and  $b > 0$ .

**Dynamic feedback linearization (DFL) [4, 5]** Exact linearization of eq. (1) can be achieved by increasing the dimension of the robot state, i.e., adding an integrator on the linear velocity input

$$v = \xi \quad \dot{\xi} = a, \quad (4)$$

and defining the new input  $(a, \omega)$  as

$$\begin{bmatrix} a \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{\xi} & \frac{\cos \theta}{\xi} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \quad (5)$$

Feedback stabilization is obtained on the linear side by choosing

$$\begin{aligned} u_1 &= \ddot{x}_d + K_{p1}(x_d - x) + K_{d1}(\dot{x}_d - \dot{x}) \\ u_2 &= \ddot{y}_d + K_{p2}(y_d - y) + K_{d2}(\dot{y}_d - \dot{y}), \end{aligned} \quad (6)$$

with  $K_{pi} > 0$ ,  $K_{di} > 0$  ( $i = 1, 2$ ). The initial state of the integrator in eq. (4) should be  $\xi_0 \neq 0$  and the desired trajectory should be persistent in order to avoid the singularity in eq. (5).

### 3.2 Parking

Let the desired configuration be, w.l.o.g., the origin of the robot state space.

**Polar coordinate control (POL) [8]** Using the polar coordinate transformation, singular at the origin,

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2} \\ \gamma &= \text{ATAN2}\{y, x\} - \theta + \pi \\ \delta &= \gamma + \theta, \end{aligned}$$

the control law is defined as

$$\begin{aligned} v &= k_1 \rho \cos \gamma \\ \omega &= k_2 \gamma + k_1 \frac{\sin \gamma \cos \gamma}{\gamma} (\gamma + k_3 \delta) \end{aligned}$$

with  $k_1$ ,  $k_2$ , and  $k_3$  all positive. This feedback law, once rewritten in the original state variables, is discontinuous at the origin.

**Dynamic feedback linearization (DFL) [9]** The DFL tracking controller (4–6) can be used also for stabilization to the origin, by setting  $\dot{x}_d = \dot{y}_d = \ddot{x}_d = \ddot{y}_d = 0$  in eq. (6). Avoidance of the singularity that may occur when the robot comes to a stop is obtained by suitable choice of the PD gains and of the initialization  $\xi_0$  (see [9]).

## 4 Experimental results

In both types of motion tasks (trajectory following and parking), the initial position and orientation of SuperMARIO is acquired through the vision system. We have implemented two operation modes, depending on whether the robot state is measured by odometry or by visual information. In the latter case, whenever the visual localization routine fails (see Sect. 2.3), odometry is used until vision-based estimation recovers.

### 4.1 Trajectory following

The reference trajectory is an eight-shaped path (see Fig. 2), with an associated rest-to-rest trapezoidal velocity profile lasting 32 s. The trajectory starts at (1, 0.8, 30), while the robot is initially at (1.7, 0.3, 90) [m, m, deg]. Figure 2 shows that, under visual feedback, DFL control has a faster transient than NTI control. Although a stable tracking is obtained, visual noise affects as expected the imposed velocities; in Fig. 3, the angular velocity command  $\omega$  obtained with odometric feedback is smoother than with visual feedback. An independent measure of the cartesian error is shown in Fig. 4 for odometric and visual feedback under DFL control. In this case, the robot traces approximately twice the eight-shaped path (60 s).

### 4.2 Parking

The robot starts at (1.1, 1.2, 135) [m, m, deg], with the origin as the desired parking goal. In Fig. 5, the robot reaches the goal in forward motion with POL control and moving only backwards with DFL control. Figure 6 shows that DFL control is faster than POL control in reducing the error to zero. Odometric and visual feedback are compared under DFL control in Fig. 7 and under POL control in Fig. 8. The performance is similar, although a final residual error is found using odometric feedback. This cartesian error is larger with POL control, essentially due to the presence of one backup maneuver, while it remains small (about 1 cm) with DFL control.

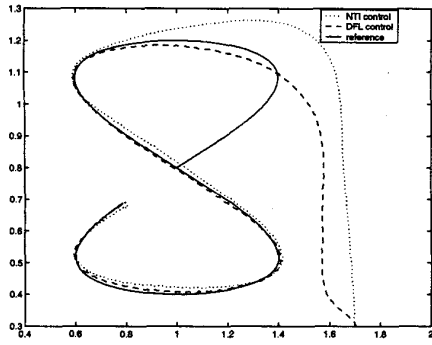


Figure 2: Trajectory following: DFL (- -) and NTI (· · ·) control using visual feedback

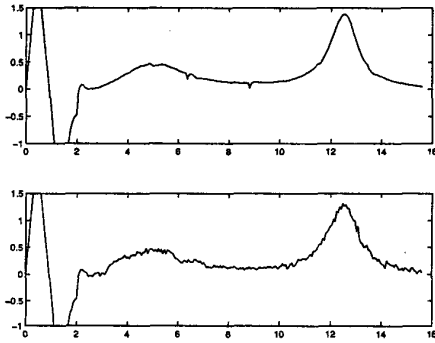


Figure 3: Trajectory following: Angular velocity  $\omega$  with odometric (above) and visual (below) feedback under DFL control

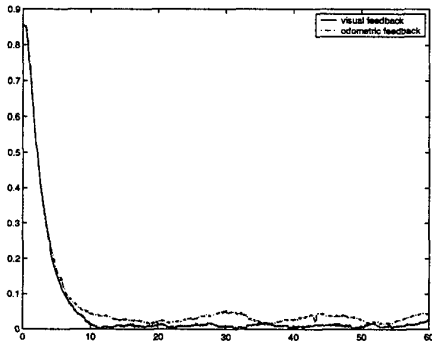


Figure 4: Trajectory following: Norm of cartesian error (in m) under DFL control using odometric (---) and visual (—) feedback

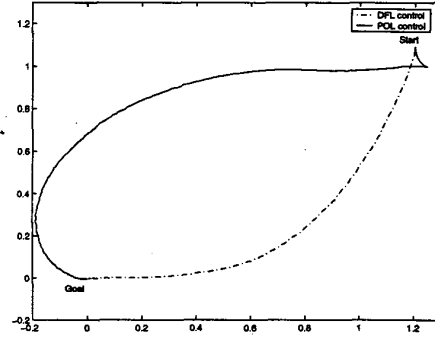


Figure 5: Parking: Cartesian path executed with DFL (---) and POL (—) control using visual feedback

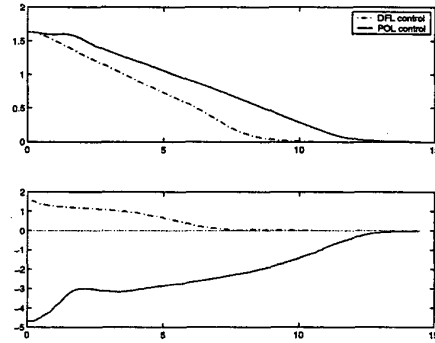


Figure 6: Parking: Norm of cartesian (top) and orientation (bottom) error with DFL (---) and POL (—) control using visual feedback

## 5 Conclusions

We have presented experimental results on visual-based motion control of the nonholonomic mobile robot SuperMARIO. Both trajectory following and parking tasks have been executed using two different feedback controllers for each case, comparing the performance obtained with odometry or visual data. The reconstruction of the robot state from visual information is based on a simple but efficient algorithm using a three-light triangle mounted on the top of the vehicle. For all controllers and motion tasks, the expected performance improvement when using visual feedback is obtained at the expense of a slight (5 ms) increase of the sampling time with respect to odometric feedback.

This work could be improved by implementing a noise reduction algorithm within the low-level image processing and performing fusion of odometric and visual information in the EKF framework. Our final objective is to achieve visual acquisition and nonholonomic motion planning in the presence of obstacles.

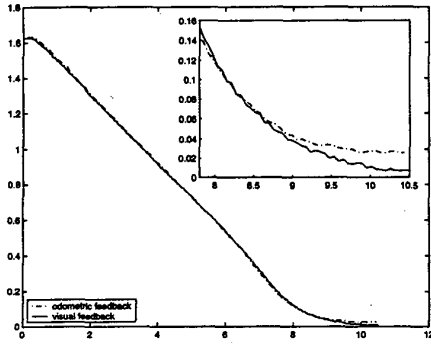


Figure 7: Parking: Norm of cartesian error under DFL control using visual (—) and odometric (---) feedback (with a zoom of the final approach)

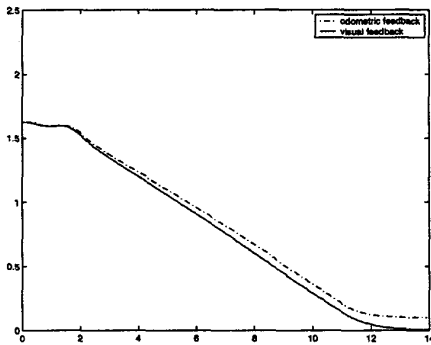


Figure 8: Parking: Norm of cartesian error under POL control using visual (—) and odometric (---) feedback

## References

- [1] G. Campion, G. Bastin, and B. d'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *IEEE Trans. on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, 1996.
- [2] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," *1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, pp. 1136–1141, 1991.
- [3] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots," *Int. J. of Robotics Research*, vol. 12, no. 1, pp. 55–64, 1993.
- [4] A. De Luca and M. D. Di Benedetto, "Control of nonholonomic systems via dynamic compensation," *Kybernetika*, vol. 29, no. 6, pp. 593–608, 1993.
- [5] B. d'Andrea-Novel, G. Bastin, and G. Campion, "Control of nonholonomic wheeled mobile robots by state feedback linearization," *Int. J. of Robotics Research*, vol. 14, no. 6, pp. 543–559, 1995.
- [6] C. Samson, "Control of chained systems. Application to path following and time-varying point-stabilization of mobile robots," *IEEE Trans. on Automatic Control*, vol. 40, no. 1, pp. 64–77, 1995.
- [7] A. Astolfi, "Discontinuous control of nonholonomic systems," *Systems & Control Lett.*, vol. 27, pp. 37–45, 1996.
- [8] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics & Automation Mag.*, vol. 2, no. 1, pp. 27–35, 1995.
- [9] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," *6th IFAC Symp. on Robot Control*, Vienna, A, pp. 397–402, 2000.
- [10] C. Canudas de Wit, H. Khenouf, C. Samson, and O. J. Sørдалen, "Nonlinear control design for mobile robots," in *Recent Trends in Mobile Robots*, Y. F. Zheng (Ed.), vol. 11, pp. 121–156, World Scientific Publisher, 1993.
- [11] A. De Luca, G. Oriolo, C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond (Ed.), LNCIS vol. 229, pp. 171–253, Springer Verlag, London, 1998.
- [12] A. De Luca, G. Oriolo, M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," in *RAMSETE. Articulated and Mobile Robots for Services and Technology*, S. Nicosia, B. Siciliano, A. Bicchi, P. Valigi (Eds.), LNCIS vol. 270, pp. 181–226, Springer Verlag, London, 2001.
- [13] B. Kwolek, T. Kapuscinski, and M. Wysocki, "Vision-based implementation of feedback control of unicycle robots," *1st Work. on Robot Motion and Control*, pp. 101–106, 1999.
- [14] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki, "Visual servoing for nonholonomic mobile robots," *1994 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 1133–1140, 1994.
- [15] K. Hashimoto and T. Noritsugu, "Visual servoing of nonholonomic cart," *1997 IEEE Int. Conf. on Robotics and Automation*, Albuquerque, NM, pp. 1719–1724, 1997.
- [16] F. Conticelli, B. Allotta, and P. K. Khosla, "Image-based visual servoing of nonholonomic mobile robots," *38th IEEE Conf. on Decision and Control*, Phoenix, AZ, pp. 3496–3501, 1999.
- [17] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.
- [18] I. T. Young, J. J. Gerbrands, and L. J. van Vliet, *Fundamentals of Image Processing*, Delft University of Technology, 1998.