May 2001

Real-Time Vision-Based Control of a Nonholonomic Mobile Robot

Aveek K. Das
University of Pennsylvania

Rafael Fierro
University of Pennsylvania

R. Vijay Kumar
University of Pennsylvania, kumar@grasp.upenn.edu

B. Southall
University of Pennsylvania

John R. Spletzer
University of Pennsylvania

See next page for additional authors

Follow this and additional works at: http://repository.upenn.edu/meam_papers

Real-Time Vision-Based Control of a Nonholonomic Mobile Robot

Abstract

This paper considers the problem of vision-based control of a nonholonomic mobile robot. We describe the design and implementation of real-time estimation and control algorithms on a car-like robot platform using a single omni-directional camera as a sensor without explicit use of odometry. We provide experimental results for each of these vision-based control objects. The algorithms are packaged as control modes and can be combined hierarchically to perform higher level tasks involving multiple robots.

Author(s)

Aveek K. Das, Rafael Fierro, R. Vijay Kumar, B. Southall, John R. Spletzer, and Camillo J. Taylor

Real-Time Vision-Based Control of a Nonholonomic Mobile Robot[1]

A. K. Das[2], R. Fierro, V. Kumar, B. Southall, J. Spletzer, and C. J. Taylor
General Robotics Automation, Sensing and Perception (GRASP) Laboratory
University of Pennsylvania
3401 Walnut Street – Suite 301C, Philadelphia, PA 19104-6228
{aveek, rfierro, kumar, southall, spletzer, cjtaylor}@grasp.cis.upenn.edu

Abstract

This paper considers the problem of vision-based control of a nonholonomic mobile robot. We describe the design and implementation of real-time estimation and control algorithms on a car-like robot platform using a single omni-directional camera as a sensor without explicit use of odometry. We provide experimental results for each of these vision-based control objects. The algorithms are packaged as control modes and can be combined hierarchically to perform higher level tasks involving multiple robots.

1 Introduction

In this paper we consider the problem of controlling the motion of a nonholonomic, car-like robot such as the one shown in Figure 1 based on the information from an onboard omni-directional camera system [3].

More specifically, in the sequel we consider three types of motion control tasks: a wall following mode where the goal is to guide the robot so that it maintains a specified orientation and distance from a boundary, a follow the leader task where the robot is instructed to maintain a prescribed position and orientation with respect to a moving target and a position regulation task where the robot is required to achieve a particular position with respect to a global reference frame.

In this work we exploit the fact that the wider field of view afforded by an omni-directional camera allows us to extract more information from the video signal and makes it possible to implement a wider range of motion strategies on the same platform.

For each of these tasks we describe the estimation schemes which are employed to recover the requisite information from the omni-directional video imagery and we discuss the control laws that accomplish the regulation task. In every case we describe how the non-holonomic constraints are accounted for in the design of both the estimation and control laws.

The problem of controlling a vehicle based on the information obtained from image data has been considered before in a number of contexts.

Horswill [11] described simple but effective vision-based behaviors that could be used to guide a robot safely through an obstacle strewn environment. Dickmanns et al [9] [17] considered the problem of controlling a motor vehicle based on the information obtained from conventional cameras mounted onboard. Ma, Kosecka and Sastry [15] looked at the problem of guiding a nonholonomic robot along a path based on visual input. Zhang and Ostrowski [18] demonstrated effective schemes for controlling a blimp system based on image measurements.

The paper is organized as follows. The experimental testbed (hardware and software) is described in section 2. Section 3 discuses the set of visual servo objects we use in our work. Experimental results are presented in section 4. Finally, some concluding remarks and future work are given in section 5.

2 The Experimental Testbed

2.1 The Mobile Robot Platform

The mobile robot we use for our experiments is shown in Figure 1. It has been constructed from a commercial radio-control truck kit. Some modifications have been made to improve shock absorption and to house an omni-directional vision system, a 2.4 GHz wireless video transmitter, and a battery pack. The robot has a servo controller on board for steering and a digital proportional speed controller for forward/backward motion. A parallel port interface, also designed in our lab, allows driving up to 8 mobile robot platforms from a single Windows NT workstation. The receiver (located at the host computer) feeds the signal to a frame grabber that is able to capture video at full frame rate (30 Hz.) for image processing. This yields a video signal in a format for viewing and recording, as well as image processing.

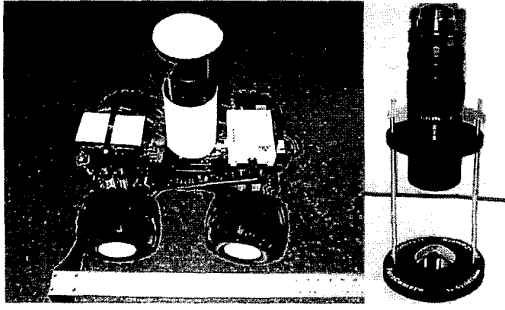[2] To whom all correspondence should be addressed

Fig. 1 The mobile robot platform with Omnicam.

2.2 The Omnicam Sensor

A *Paracamera* manufactured by Cyclovision consists of a parabolic mirror and a lens assembly mounted on a Pulnix ½" remote head color CCD camera.

This omni-directional sensor provides us with a 360° field of view. There exists a simple geometric mapping from the image plane to the ground plane or any other plane of interest [3] due to the special parabolic shape of the reflecting surface. This is useful for doing navigation and control.

2.3 Vision Algorithms

Color feature extraction: Pixels corresponding to the object of interest are extracted from the image using a statistical color model in YUV space that provides some robustness to variations in viewing and illumination conditions. The color detection algorithm is applied to the whole image (this process runs at 15-20 Hz for upto three colors). If the object of interest is detected, then we switch into target tracking mode.
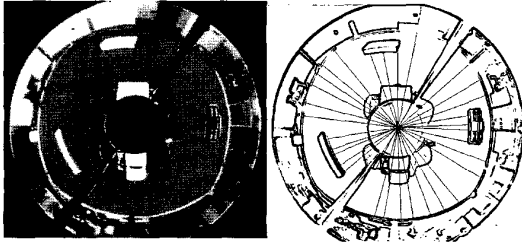

Fig. 2 Omni-directional image and associated edge image and range map.

Range mapping: A Sobel gradient was applied to the original omni-directional image. The resulting edges in the image were assumed to be features of interest. By assuming a ground plane constraint, the distance to the nearest feature in the sector of interest was determined from the its relative elevation angle to the mirror. This provides a range map to all obstacles at frame rate. A similar approach to determining range from image data was employed by Horswill [11].

We have integrated these vision algorithms in a multi-threading software architecture for implementation on the

hardware platforms for basic functionality like wall following, obstacle avoidance, leader following and so on.

3 Vision-Based Control Objects

3.1 Wall Follower

The wall follower operates by taking inputs from 2 "sensors" – a wall detector and an obstacle detector. These are implemented as separately threaded rangemaps with different sectors of interest (160-200° and 50-130°, respectively). The wall detector extracts points from each of its nine 5-degree sectors. A line is fit to these points using Random Sample Consensus (RANSAC), which yields a fit robust to outliers [10]. From this, the relative position and orientation of the wall can be calculated. The obstacle detector picks up features in its 80° field of view. Since the position and orientation relative to the wall are known, the detector is able to discriminate which features are actually the wall, and which are truly obstacles that must be avoided. I/O feedback linearization techniques are used to design a PD controller to regulate the distance of the vehicle to the wall.

Wall following can be considered as a particular case of path following. Thus, the kinematics in terms of the path variables become [7]

$$\dot{s} = u_1 \cos\theta_p, \qquad \dot{d} = u_1 \sin\theta_p$$

$$\dot{\theta}_p = u_1 \frac{\tan\phi}{l}, \qquad \dot{\phi} = u_2 \tag{2}$$

$$\theta_t = \frac{\pi}{2}$$

$$\theta_p = \theta - \theta_t$$

$$\theta_p = \theta - \frac{\pi}{2}$$
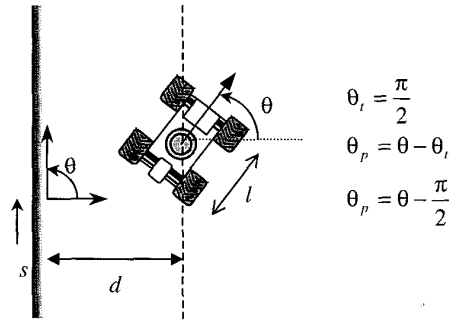
Fig. 3 Wall following.

Assuming the robot is to follow the wall with a piecewise constant velocity $v_1(t)$, the system output and its time derivative are given by

$$z(t) = h(x) = d(t), \qquad \dot{z} = \dot{d}(t) \tag{3}$$

Then, we obtain

$$\ddot{d} = u_1 \dot{\theta}_p \cos\theta_p = \frac{u_1^2}{l} \cos\theta_p \tan\phi \equiv r \tag{4}$$

If the desired distance to the wall is d_0, r is given by

$$r = \ddot{d}_0 + k_v(\dot{d}_0 - \dot{d}) + k_p(d_0 - d) \tag{5}$$

After some work, we have

$$v_2 = \tan^{-1}\left[\frac{l}{u_1^2 \cos\theta_p}(k_p(d_0 - d) - k_v u_1 \sin\theta_p)\right] \tag{6}$$

1715

where $v_2(t)$ is the steering command, $u_l(t)$ is the linear velocity, and k_p, k_v are positive design controller gains. Usually, we may want a critically damping behavior i.e.,

$$k_v = 2\sqrt{k_p} \ .$$

3.2 Velocity Estimator

The *Leader Follower* object described next, requires reliable estimation of the linear velocity $v_i(t)$ and angular velocity $\omega_i(t)$ of the leader mobile robot R_i, and relative orientation $(\theta_i - \theta_j)$. The velocity *estimator* algorithm is based on an extended Kalman filter [14], [16]. It uses the omni-directional vision system to determine the range ρ_{ij} and the bearing β_{ij} of the observed leader R_i, see Figure 4. In addition, the filter requires a sensor model and the relative kinematic equations [8] of the leader R_i and follower R_j. The image processing algorithms provide the following observations

Range: $\qquad \rho_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \qquad$ (7)

Bearing: $\qquad \beta_{ij} = \dfrac{\pi}{2} + a\tan 2(y_i - y_j, x_i - x_j) - \theta_j \qquad$ (8)

The closed-loop omni-directional vision system for leader tracking is depicted in Figure 6. Let us define

$$
\begin{aligned}
x_{ij} &\equiv x_i - x_j \\
y_{ij} &\equiv y_i - y_j \\
\alpha_{ij} &\equiv \beta_{ij} + \theta_j - \theta_i
\end{aligned}
\qquad (9)
$$

Then, $\qquad \dot{\rho}_{ij} = \dfrac{x_{ij}\dot{x}_{ij} + y_{ij}\dot{y}_{ij}}{\rho_{ij}} \qquad$ (10)

and $\qquad \dot{\beta}_{ij} = \dfrac{x_{ij}\dot{y}_{ij} - y_{ij}\dot{x}_{ij}}{\rho_{ij}^2} - \dot{\theta}_j \qquad$ (11)
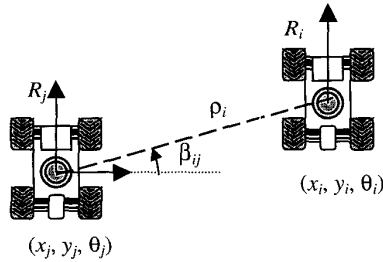

Fig. 4 Robot configuration for velocity estimation.

The state vector to be estimated is given by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, \mathbf{w})$$

$$
\begin{bmatrix} \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \\ \dot{\rho}_{ij} \\ \dot{\beta}_{ij} \\ \dot{\theta}_j \end{bmatrix} =
\begin{bmatrix} \omega_i \\ 0 \\ 0 \\ v_i \sin\alpha_{ij} - v_j \sin\beta_{ij} \\ \dfrac{v_i \cos\alpha_{ij} - v_j \cos\beta_{ij}}{\rho_{ij}} - \omega_j \\ \omega_j \end{bmatrix} + \mathbf{w}(t) \qquad (12)
$$

where $\mathbf{w}(t)$ is the process noise, assuming $\dot{v}_i \approx 0, \dot{w}_i \approx 0..$ The system output with sensor noise is given by

$$\mathbf{z}(t) = \mathbf{h}(\mathbf{x}) + \eta(t) = \begin{bmatrix} \rho_{ij} & \beta_{ij} \end{bmatrix}^T \qquad (13)$$

The discrete system becomes

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) + \mathbf{w}(k), \quad \mathbf{w}(k) \sim N(\mathbf{0}, \mathbf{Q}(k))$$

$$
\mathbf{F}(\mathbf{x}(k), \mathbf{u}(k)) =
\begin{bmatrix}
\theta_i + \omega_i \Delta T \\
v_i \\
\omega_i \\
\rho_{ij} + (v_i \sin\alpha_{ij} - v_j \sin\beta_{ij})\Delta T \\
\beta_{ij} + \Delta T \left[\dfrac{v_i \cos\alpha_{ij} - v_j \cos\beta_{ij}}{\rho_{ij}(k)} - \omega_j \right] \\
\theta_j + \omega_j \Delta T
\end{bmatrix}
\quad (14)
$$

where $\mathbf{F}(\mathbf{x}(k), \mathbf{u}(k))$ is the nonlinear state transition function. The input vector is given by $\mathbf{u} = \begin{bmatrix} v_j & \omega_j \end{bmatrix}^T$. $\mathbf{w}(k)$ is a noise source assumed to be zero-mean Gaussian with covariance $\mathbf{Q}(k)$. ΔT is the sampling interval (~ 50 ms). The discrete (observation) output is given by

$$\mathbf{z}(k) = \mathbf{h}(\mathbf{x}(k)) + \eta(k), \qquad \eta(k) \sim N(\mathbf{0}, \mathbf{R}(k)) \qquad (15)$$

$\mathbf{R}(k)$ is experimentally determined. The goal of the EKF algorithm is to estimate $\hat{\mathbf{x}}(k+1|k+1)$ and its covariance $\mathbf{P}(k+1|k+1)$ given, $\hat{\mathbf{x}}(k|k)$, $\mathbf{P}(k|k)$ at time k, and the current observation $Z(k+1)$. We use a standard estimation algorithm, see for instance [13], [15], where the observation vector and measurement prediction are given by

$$\mathbf{z}(k+1) = \begin{bmatrix} \rho(k+1) & \beta(k+1) \end{bmatrix}^T \qquad (16)$$

$$\hat{\mathbf{z}}(k+1) = \mathbf{H}\hat{\mathbf{x}}(k+1|k) \text{ with } \mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (17)$$

3.3 Leader Follower

Details of this controller are given in [8]. The *Velocity Estimator* object provides the follower with necessary information about the velocity of the leader for feed-forward control. This eliminates the need for explicit communication. The basic structure for this object is shown in Fig 6.
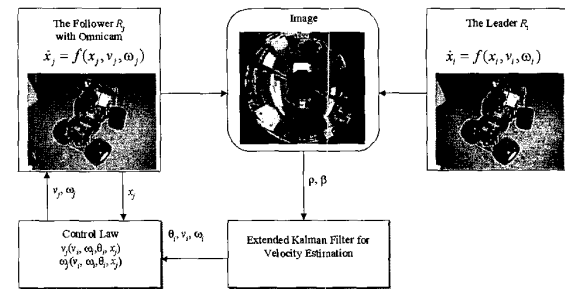

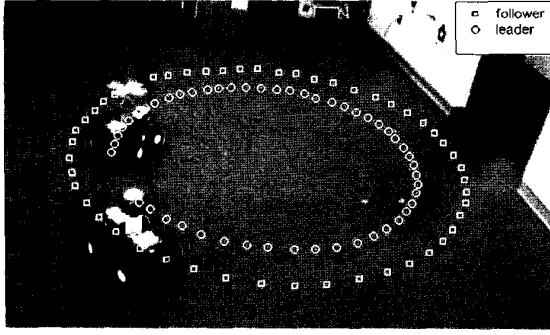**Fig. 6** Leader Follower structure.

Fig. 7 Overhead view of leader following. These are actual data points collected from a single run.

3.4 Localization

We have implemented a localization algorithm for our mobile robot. The algorithm also employs an extended Kalman filter (EKF) to match landmark observations to an a priori map of landmark locations [14]. The *Localizer* object uses a *BlobExtractor* sensor to determine the range and the bearing of an observed landmark. If the observed landmark is successfully matched, it will be used to update the vehicle position and orientation. Figure 8 depicts a typical image used for localization.

The algorithm is similar to the one used for velocity estimation. The main differences are explained bellow. The *kinematic model* of the mobile robot, in this case, is given by

$$
\begin{aligned}
\dot{x} &= u_1 \cos\theta \\
\dot{y} &= u_1 \sin\theta \\
\dot{\theta} &= \frac{u_1 \tan\phi}{l} \\
\dot{\phi} &= \lambda_\phi (u_2 - \phi)
\end{aligned}
\tag{18}
$$

where l is the body length, u_2 is the steering command, $|\phi| < 70°$ is the steering angle, and $\lambda_\phi \approx 4$ s^{-1} is a parameter that depends on the steering servo time constant and wheel-ground friction. The control vector is given by $\mathbf{u} = [u_1 \ u_2]^T$.
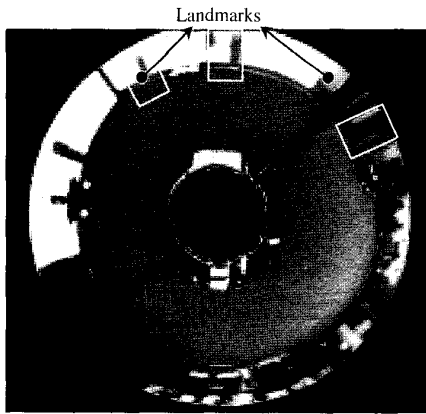

Landmarks

Fig. 8 Image used for localization.

The linearized kinematics become

$$
\nabla\mathbf{F} \equiv \mathbf{A} :=
\begin{bmatrix}
1 & 0 & -u_1(k)\sin(\hat{\theta}(k\,|\,k))\Delta T & 0 \\
0 & 1 & u_1(k)\cos(\hat{\theta}(k\,|\,k))\Delta T & 0 \\
0 & 0 & 1 & \dfrac{u_1(k)\sec^2(\phi(k))}{l}\Delta T \\
0 & 0 & 0 & 1 - \lambda_\phi\Delta T
\end{bmatrix}
\tag{19}
$$

We use a map that consists of n_p landmarks. The landmark position is given by $\mathbf{p}_i = (p_x, p_y)$, and the *output function* becomes

$$
\mathbf{h}_i(\mathbf{p}_i, \mathbf{x}(k)) =
\begin{bmatrix}
\sqrt{(p_x - x(k))^2 + (p_y - y(k))^2} \\
\tan^{-1}\left(\dfrac{p_y - y(k)}{p_x - x(k)}\right) - \theta(k)
\end{bmatrix}
\tag{20}
$$

When the image processing is completed, we obtain a set of observations $Z(k) = \{\mathbf{z}_j(k) \mid j = 1,2,....,n_0\}$.

To compute the variance of the innovation we need the observation Jacobian for each prediction

$$
\nabla\mathbf{h}_i = \frac{1}{r_P^2}
\begin{bmatrix}
(\hat{x}(k+1|k) - p_x)r_P & (\hat{y}(k+1|k) - p_y)r_P & 0 & 0 \\
p_y - \hat{y}(k+1|k) & \hat{x}(k+1|k) - p_x & -r_P^2 & 0
\end{bmatrix}
\tag{21}
$$

$$
r_P \equiv \sqrt{(p_x - \hat{x}(k+1|k))^2 + (p_y - \hat{y}(k+1|k))^2}
\tag{22}
$$

where r_P is the distance from the sensor to the ith detected landmark.

The following expression is used to validate and match each sensor observation. The innovation is $\mathbf{v}_{ij}$ and $\mathbf{S}_{ij}$ is its variance.

$$
\mathbf{v}_{ij}(k+1)\mathbf{S}_{ij}^{-1}(k+1)\mathbf{v}_{ij}^T(k+1) \leq \chi^2
\tag{23}
$$

Measurements not satisfying the above criteria are ignored for localization. Finally, the Kalman gain is computed and the robot's position vector and the associated variance are updated over all matched landmarks.

3.5 Go To Goal Controller

The *GoToGoal* controller is derived from the *leader follower* controller [8] considering the goal configuration (x_g, y_g, θ_g) to be a virtual leader robot with zero linear and angular velocities. The *kinematic model* of the robot is again given by (18). The final orientation will depend on the initial conditions (choice of desired ψ) Then, the controller takes the form –

$$
\begin{aligned}
v &= l_g \cos\gamma - \dot{\psi}_g l_g \sin\gamma \\
\omega &= \frac{1}{d}[l_g \sin\gamma + \dot{\psi}_g l_g \cos\gamma]
\end{aligned}
\tag{24}
$$

where, $\quad \gamma = \theta_g + \psi_g - \hat{\theta}$

The *Localizer* gives estimates of self configuration $(\hat{x}, \hat{y}, \hat{\theta})$, which is then used to calculate position of the control point (with offset d from axle).

$$\overline{x} = \hat{x} + d \cos \hat{\theta}$$
$$\overline{y} = \hat{y} + d \sin \hat{\theta}$$

(25)

Now,

$$\psi_g = \pi - \arctan(\overline{y} - y_g, \overline{x} - x_g) - \theta_g$$
$$l_g = \sqrt{(x_g - \overline{x})^2 + (y_g - \overline{y})^2}$$

(26)

The linearized system for (20) becomes

$$\dot{l}_g = -k_1 l_g$$
$$\dot{\psi}_g = k_2 (\psi_g^d - \psi_g)$$
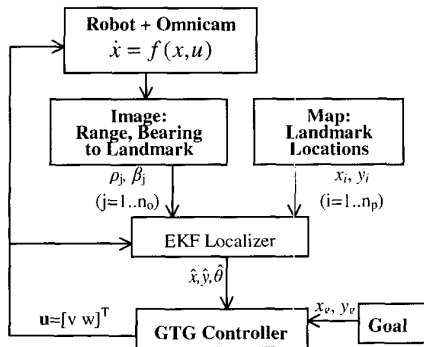
(27)

Fig. 10 The GoToGoal structure

4 Experimental Results

4.1 Wall Following

In this experiment we start the robot off near a wall with an obstacle (box) midway along the length of the wall. The aim of the wall follower is to maintain an offset of 30cm (12in) from the wall unless it encounters an obstacle, in which case it switches to obstacle avoidance until it sees a clear wall along-side again. The results are shown below with the ground truth from the overhead camera and the relevant mode switches. The units on the x and y axes are inches.
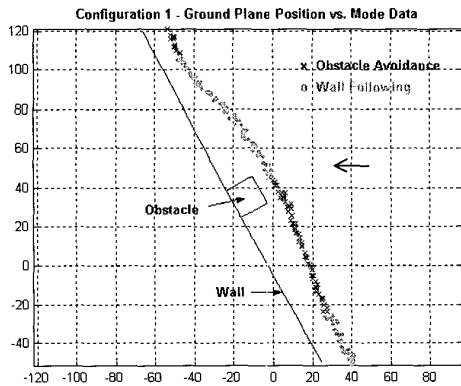

Fig. 11 The Wall Follower showing both modes

4.2 Velocity Estimator and Leader Follower

The omni-directional vision system provides the range and bearing of the observed leader robot. This information is fed to the velocity estimator. Control velocities for the follower robot are computed and sent to the driving and steering servos.
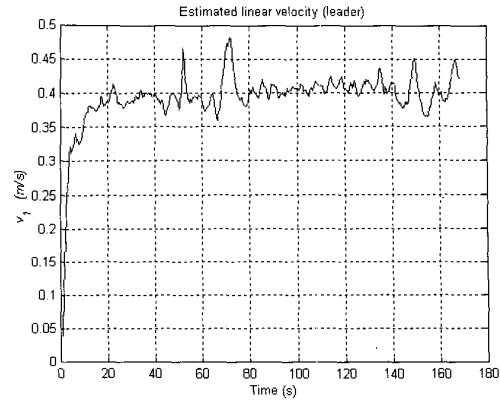

Fig. 12a Linear velocity estimation of leader from follower

In this experiment the leader is tracking a circular path, and the follower is to follow the leader with desired separation and bearing 0.6 m and 180° respectively. To study the robustness of our system, we perturb it by holding the follower (see figure 12 at t = 65 s.) or blocking its line of sight. In both cases the system was able to recover and continue its task.
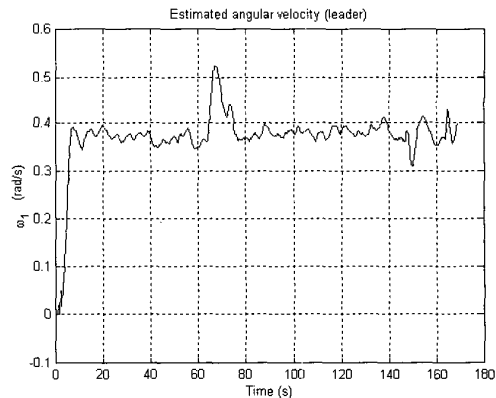

Fig. 12b Angular velocity estimation of leader from follower

Finally, it is worth noting that control velocities are computed using the leader's velocity estimates i.e., there is no explicit communication between the follower and the reference robot.

4.3 Mobile Robot Localization

In this experiment we let the robot trace an open loop circular trajectory in a measured area with fixed landmarks. Due to the windowed tracking approach, the robot usually sees very few landmarks. The overhead

1718

camera gives us an idea of the actual ground trajectory of the robot. The average error is ~2 cm. This is particularly challenging as we use a simplified kinematic model, and we lack odometry.
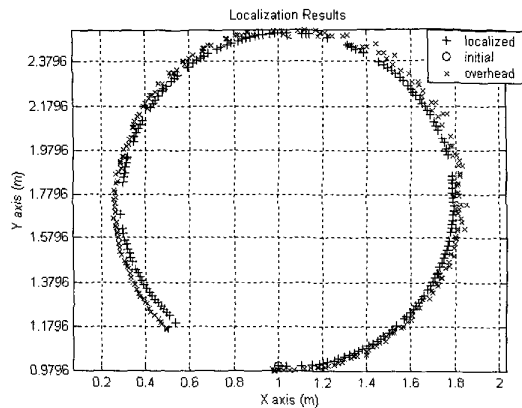

Fig. 13 Localization results for a circular trajectory.

4.4 Goal Acquisition

We are in the process of implementing this controller on the robots. The following are the results from simulations using this controller. In figure 14 the virtual leader robot
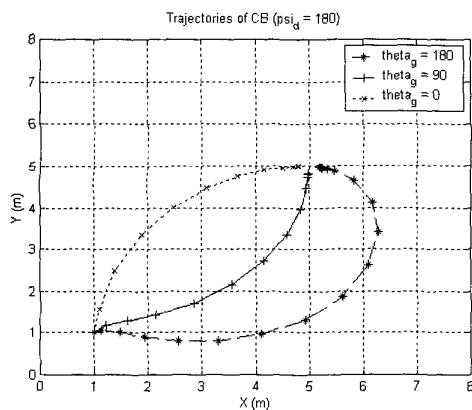

Fig. 14 Trajectories of robot for go to goal (5,5) starting from (1,1) with orientation 180°.

at the goal is at orientations of 0, 90 and 180°, while the desired bearing for the follower is 180° from the heading of the leader.

5 Conclusions

In this paper, we have presented a framework for real-time control using an omni-directional vision system. Experimental results verify the validity of our approach. Velocity estimation and localization techniques based on an EKF have been integrated in the closed loop system. The experimental results are being extended to more complex scenarios. Moreover, sophisticated visual control modes are being implemented in our nonholonomic mobile platforms. These include planning, collaborative mapping, and formation keeping.

References

1. A. Adam, E. Rivlin, and I. Shimchoni, "Computing the sensory uncertainty field of a vision-based localization sensor," *Proc. IEEE Int. Conf. Robot & Automat.*, San Francisco, CA, April 2000, pp. 2993-2999.

2. M. D. Adams, *Sensor Modeling, Design and Data Processing for Autonomous Navigation*, Singapore: World Scientific, 1999.

3. S. Baker and S. Nayar, "A Theory of Catadioptric Image Formation", ICCV '97, pp. 35-42, January 1998.

4. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, Collaborative multi-robot exploration, *Proc. IEEE Int. Conf. Robot & Automat.*, San Francisco, CA, April 2000, pp. 476-481.

5. F. Conticelli, D. Prattichizzo, F. Guidi, and A. Bicchi, "Vision-based dynamic estimation and set-point stabilization of nonholonomic vehicles," *Proc. IEEE Int. Conf. Robot & Automat.*, San Francisco, CA, April 2000, pp. 2771-2776.

6. P. I. Corke and S. A. Hutchinson, "Real-time vision, tracking and control," *Proc. IEEE Int. Conf. Robot & Automat.*, San Francisco, CA, April 2000, pp. 622-629.

7. A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*, J.-P. Laumond (ed.), London: Springer-Verlag, 1998, pp. 171-253.

8. J. Desai, J. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," *Proc. IEEE Int. Conf. Robot Autom.*, Leuven, Belgium, May 1998, pp. 2864-2869.

9. E. D. Dickmanns and A. Zapp, "Guiding Land Vehicles along Roadways by Computer Vision", *Congres Automatique*, 1985, 233-244.

10. M. Fischler, and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applciations to Image Analysis and Automated Cartography", *Communcations of the ACM*, June 1991.

11. Horswill, I., "Polly: A Vision-Based Artificial Agent", *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, Washington DC, MIT Press, July 11-15, 1993.

12. S. A. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651-670, 1996.

13. H. K. Khalil, *Nonlinear Systems*, Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1996.

14. J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Sensing for Mobile Robot Navigation*, Boston: Kluwer Academic Publishers, 1992.

15. Y. Ma, J. Košecká, and S. Sastry, "Vision guided navigation for nonholonomic mobile robot," *IEEE Trans. Robot. Automat.*, vol. 15, no. 3, pp. 521-536, June 1999.

16. G. M. Siouris, *An Engineering Approach to Optimal Control and Estimation Theory*, New York: John Wiley & Sons, Inc., 1996.

17. C. J. Taylor, J. Košecká, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," The Int. J. Robot. Research, vol. 18, no. 5, pp. 42-453, 1999.

18. Hong Zhang and James P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned Blimp", *Proc. Int. Conf. on Robotics and Automation*, Detroit, MI, pp. 618-623, May 1999.