Vision Based Teleoperated Cellular Robots

Carlos Carreira and José Santos-Victor

Instituto Superior Técnico & Instituto de Sistemas e Robótica
Av. Rovisco Pais - Lisboa Portugal
e.mail : {cc,jasv}@isr.ist.utl.pt

Abstract. This paper describes the development and design of a vision-based teleoperated cellular robot. Cellular robots can find numerous application in industrial problems where simple inexpensive robots can be used to perform different tasks that involve covering a large working space. As a methodology, the robots are controlled based on the visual input from one or more cameras that observer the working area. As a result, a robust control of the robot trajectory is achieved and does not depend on the camera calibration. As an example, we present a system, where the remote user can simply specify a target point in the image to indicate the robot final position.
We describe the complete system at various levels: the visual information processing, the robot characteristics and the closed loop control system design. Preliminary results are presented and discussed.
In our opinion, such a system may have a wide spectrum of applications in industrial robotics and may also serve as an educational testbed for advanced students in the fields of vision, robotics and control.

1 Introduction

The remote surveillance of a working space is a problem often found in industrial applications (surveillance, inspection, mine localization, etc). We address this problem by designing simple and inexpensive cellular robots that can individually or cooperativelly operate in a large area or an hostile or remote environment [8].

A major constraint is the low cost of each robot. In the approach we follow, each robot has a very simple and inexpensive design and communicates via radio to a central processing unit, where all the costly system components are installed. We assume that a surveillance camera is available that allows the remote user to visualize the scene where the robots perform various tasks. The user can specify commands using a simple interface like indicating the desired end position of each robot by pointing an image area. The robot is controlled through visual feedback [2] and the user specifies the desired end position or trajectory exclusively in image coordinates. All the problems related to the camera calibration (intrinsic and extrinsic parameters) are embedded in the control system, in a way transparent to the user. This general setup is illustrated in Figure 1.

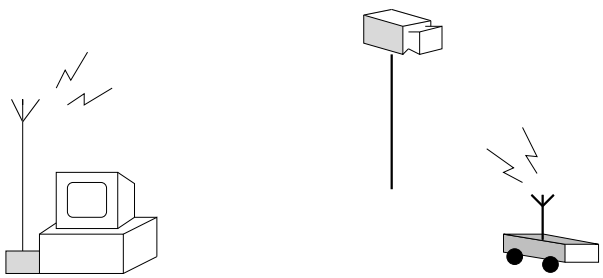

Fig. 1.: Overall system setup

The central computer runs the tasks of determining the robot position and orientation in the image coordinates and determines the proper control signals to be sent to each robot. The communication is assured via serial port, where a simple protocol was developed to send the control commands.

Currently, the robot is assembled from LEGO parts and contains a CPU-board consisting of the radio receiver, decoding logic and PWM generation for the differential control of the robot. Each robot is identified by an 4 bit address and hence, we can control up to 16 cellular vehicles.

The system we present consists of a single robot control via a radio link from a PC. The robot has no on-board sensors, but can be easilly expanded according to the needs. In the future we plan to add proximity sensors and we are currently developing a digital camera that can be used to send images of the inspected region to the remote operator.

The visual tracking runs at about 8Hz, without using any special image processing hardware. We have carefully addressed the problem of designing a digital control for the system, which is detailed in this paper. Apart from the industrial applications, this system offers a testbed to study the cooperation of multiple robots and can serve as a valuable educational kit for students in the areas of robotics, computer vision and control.

2 Visual Sensing

In order to control the trajectory of a robot, we have first to relate the robot position on the pavement and the robot image position observed by the remote user.

2.1 Determining the robot ground plane coordinates

It is well known that under perspective projection the 3D coordinates of a scene point and the corresponding image projection are related by the following equation [3].

$$\tilde{m} = \tilde{P}\tilde{M}$$

$$\begin{bmatrix} \lambda x_p \\ \lambda y_p \\ \lambda \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_o & 0 \\ 0 & fk_v & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where $\tilde{m}$ denotes the image position in projective coordinates; $\tilde{M}$ designates the 3D point, expressed in projective coordinates, relative to a world coordinate frame. The matrix $\tilde{P}$ is a 3×4 that depends on the camera intrinsic parameters (k_u, k_v, u_0, v_o, f) and the camera position relative to the work frame (r_{ij}, t_x, t_y, t_z).

As, in our case, the vehicle is moving on a plane π, we can choose the world coordinate frame such that the plane equation is given by $Z = 0$ and as a result each image point will correspond univoquely to a certain point on the plane π, according to :

$$\tilde{m} = \tilde{P}_p\tilde{M}$$

$$\begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2)$$

where X, Y denote the vehicle position on the ground plane.

Equation (2) expresses a plane-to-plane projective transformation. The transformation matrix $\tilde{P}_p$ has 8 degrees of freedom (up to a scale factor) and can be determined from four image points with known coordinates on the plane π. Once $\tilde{P}_p$ has been estimated, it can be used to convert the coordinates from the image plane to the ground plane and vice-versa [3].

2.2 Visual Tracking : the $\alpha - \beta$ tracker

To control the robot motion, we must sense its position and heading direction over time, using the camera covering the operation work space. A strong requirement is to obtain these measurements at frequency allowing the closed loop control strategy [1].

Since the camera is fixed, the vehicle image position is estimated by computing the differences between two consecutive images and computing the centroid of the detected region. In addition, an adaptive window is used and this operation is only performed in a small neighborhood of the actual vehicle position. Whenever the vehicle is lost by the tracking system, the adaptive window is progressively enlarged thus covering a large

part of the visual field. In general, the tracking system runs at a frequency of about $8Hz$ and is robust against multiple moving objects since the processing is done locally.

The visual process provides estimates of the vehicle image position over time. These measurements have to be used to estimate the vehicle heading direction. Estimating the trajectory direction from a set of densely sampled position measurements is a very noisy process, mainly because for small image displacements the uncertainty on the angular orientation is extremely high due to the discrete nature of the image coordinates. The solution is to use an adequate filter to estimate the trajectory heading from position observations, or equivalently the vehicle instanteneous velocity.

We model the target to track as a dynamical system where the state state vector is composed both of the vehicle position and velocity, whereas the observations consist on the target position over time:[1]

$$X(k) = [x \quad \dot{x}]^T \qquad\qquad Z(k) = x \tag{3}$$

The state-space model of the dynamical system, assuming constant velocity is given by:

$$X(k+1) = \phi X(k) + \psi w(k)$$

$$Z(k) = C^T X(k) + \eta(k) = [1 \quad 0]X(k) + \eta(k) \tag{4}$$

where η is the measurement uncertainty noise, assumed to be a zero-mean white stationary process; $w(k)$ is the unknown target maneuver; ϕ and ψ are the state and maneuver-state transition matrices:

$$\phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad\qquad \psi = \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}$$

where T denotes the sampling period. The problem then is to estimate the state vector using the set of observations available up to the current sampling instant:

$$\hat{X}(k|k) = f(\{Z(n) : n = 0, 1, ..., k\}, \phi, \psi)$$

that minimizes the covariance matrix, P of state estimation error, $\tilde{X(k)}$:

$$\tilde{X}(k) = X(k) - \hat{X}(k|k)$$
$$P(k|k) = E\{\tilde{X}\tilde{X}^T\}$$

where we assume that the estimation error is zero mean. In the ultimate case, the error covariance matrix is a function of the estimation process itself and of the measurements and maneuvering noise variations, σ_η^2 and σ_w^2:

$$P(k|k) = P(f, \sigma_\eta^2, \sigma_w^2).$$

Since the system is linear and the noise processes are white, the optimal mean-squared-error (MSE) filtering is the Kalman filter [4]. We can now use the *tracking index* introduced in [6] that characterizes the general tracking problem and further simplifies the optimal tracking solution.

The tracking index, λ, is defined as a ratio between the position uncertainty due to the target maneuvers, and the sensor measurement noise and it is proportinal to the tracking period:

$$\lambda \triangleq T^2 \frac{\sigma_w^2}{\sigma_\eta^2}$$

The tracking index completeley describes the tracking solution since once it is defined, the optimal filter is implicitely defined, together with the tracking performance criterion:

$$P_{opt}(k|k) = P_{opt}(k, \lambda).$$

[1] For the sake of simplicity we represent the vehicle position by x, even if it actually consists of both horizontal and vertical coordinates. Hence the state is a 4-tuple vector and the observations a 2-tuple vector. Since motion model along the x and y directions is decoupled, we can apply the $\alpha - \beta$ filter independently in both directions.

Now that the system model has been defined and the tracking index introduced, the optimal steady-state filter is given by:

$$\textbf{Prediction :} \qquad \hat{X}(k|k-1) = \phi \hat{X}(k-1|k-1)$$

$$\textbf{Filtering :} \qquad \hat{X}(k|k) = \hat{X}(k-1|k-1) + K\left[Z(k) - C\hat{X}(k|k-1)\right]$$

where K is the steady-state gain vector given by:

$$K = \begin{bmatrix} \alpha & \frac{\beta}{T} \end{bmatrix}^T$$

Given the tracking index, the optimal gains can be computed in a straight forward manner, yielding:

$$\alpha = -\frac{\lambda^2 + 8\lambda - (\lambda+4)\sqrt{\lambda^2+8\lambda}}{8}$$

$$\beta = \frac{\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2+8\lambda}}{4}$$

$$\lambda = \frac{\beta^2}{1-\alpha}$$

Even in the cases when the characteristics of maneuverability of the vehicle are not known a priori, a smooth trajectory can be obtained using low values for λ. In our case, since the sampling period is constant and the position measurement noise is in the order of magnitude of one pixel, we can change the value of λ, thus changing the filter bandwith.

3 Robot Kinematics and Dynamics

To control the robots, one has first to define the various coordinates systems attached to the robot - kinematics - and provide a dynamic model to be used when designing the control system.

3.1 Kinematics

The vehicle was designed according to a simple differential architecture where two DC motors drive directly the left and right wheels, as shown in Figure 2.

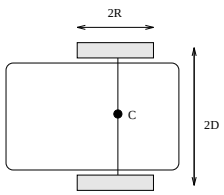

Fig. 2.: Vehicle Kinematics

A simplified kinematic model can be obtained by considering a coordinate frame attached to the mid-point, C in the wheels axis, as shown in Figure 2. The linear and angular velocities of C, V_c and ω_c, can be expressed as a function of the angular velocities of the left and right wheels ω_L and ω_R; the distance between the wheels, $2D$, and the diameter of the wheels $2R$, according to :

$$\begin{bmatrix} V_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ -R/D & R/D \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{5}$$

This jacobian matrix allows us then to determine the velocity of the vehicle reference point, as a function of the velocity of the wheels, and vice-versa (using the inverse jacobian).

3.2 Dynamics

We have now to consider the dynamic model of the vehicle. We consider that DC motors are being used and that the electrical pole can be discarded in comparison with the dominant mechanical pole. The transfer function between the motor shaft position and the input voltage is well approximated by a first-order dynamic system[7] :

$$\dot{\Theta}(s) = \frac{K_\omega \tau}{s+\tau} U(s) \tag{6}$$

where τ is the mechanical pole, and K_ω is a gain factor.

If the kinematic model is now used, we can obtain the transfer functions between the left/right motor input voltages and the angular and linear position of the vehicle reference point :

$$\dot{\Theta}_C(s) = \frac{R}{D}\frac{K_\omega \tau}{s+\tau} U_{diff}(s), \qquad U_{diff}(s) = U_R(s) - U_L(s)$$

$$V_C(s) = \frac{R}{2}\frac{K_\omega \tau}{s+\tau} U_{comm}(s), \qquad U_{comm}(s) = U_R(s) + U_L(s) \tag{7}$$

In conclusion, by using the expressing the central point of the wheels axis as the vehicle reference point, we obtain two *decoupled* single-input-single-output systems that can be controlled separately. We can now design a control system for the vehicle heading (actuating on the differential voltage) and linear speed (actuating on the common voltage).

4 Control System Design

For now we consider only the design of the heading control system. The control strategy consists in dynamically orienting the vehicle heading towards the desired final point. Along the path, the vehicle will move with a constant cruise speed. Alternatively, we could also control the linear speed as a function of the distance to the final point, thus slowing down the robot when approaching the goal position.

We will use a digital control system [7] that reads the vehicle position and orientation data sensed by the vision system and generates the appropriate differential voltages to control the trajectory heading. The overall control chain shown in Figure 3, including an impulsive sampling block, together with the zero-order-hold circuit at the controller output. where $G(s)$ denotes the transfer function between the vehicle heading

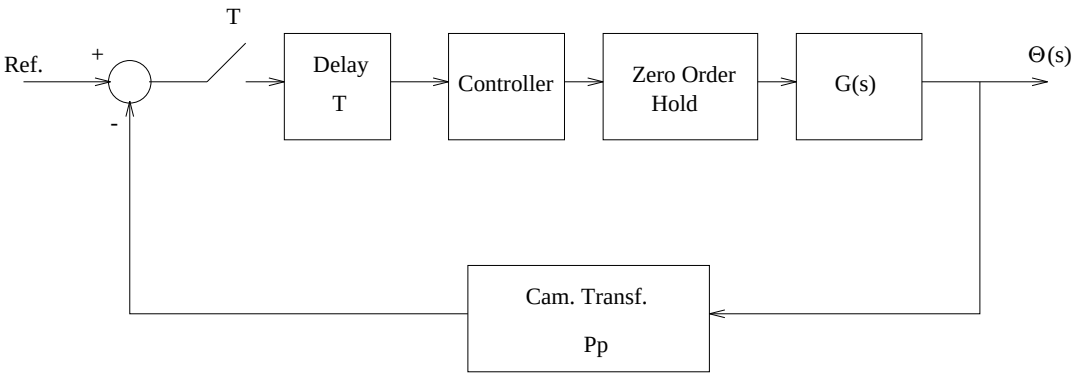

Fig. 3.: Control system

and the differential voltage U_{diff}:

$$G(s) = \frac{R}{D}\frac{K_\omega \tau}{s(s+\tau)} \tag{8}$$

The impulsive transfer function including the Zero Order Hold circuit and the process $G(S)$ can be computed from :

$$\begin{aligned} G_{eq}(Z) &= \mathcal{TZ}\left\{\frac{1-e^{-Ts}}{s}\cdot\frac{R}{D}\frac{K_\omega\tau}{s(s+\tau)}\right\} \\ &= (1-Z^{-1})\mathcal{TZ}\left\{\frac{K_\omega\tau}{s^2(s+\tau)}\right\} \\ &= \frac{RK_\omega}{D\tau}\frac{AZ+B}{(Z-1)(Z-e^{-\tau T})} \end{aligned} \tag{9}$$

where we have defined

$$A = e^{-\tau T} + \tau T - 1 \qquad B = 1-(1+\tau T)e^{-\tau T}$$

One has still to consider the delay of one sampling period introduced by the image processing. Therefore one extra sample delay Z^{-1} has to be considered. Thus, the discrete open loop transfer function is the following:

$$G_{OL}(Z) = \frac{RK_\omega}{D\tau}\frac{AZ+B}{Z(Z-1)(Z-e^{-\tau T})} \tag{10}$$

The root-locus corresponding to this open-loop discrete transfer function is shown as the leftmost plot in Figure 4. Hence, for a certain range of the gain, the system is stable and becomes unstable for large gains. An

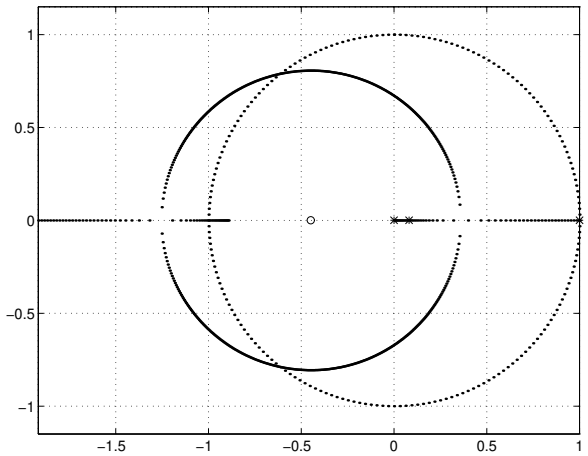

Fig. 4.: System Root-Locus when a proportinal controller is used.

additional conclusion is that inserting an integral action in the controller would make the system unstable as there already is an integrator in the loop. Then a simple PD (proportional derivative controller) is a suitable choice. This controller has a pole at the origin of the $\mathcal{Z}$ plane and a zero on the right hand side of the same plane:

$$u_{PD}[n] = K_p[e_{[n]} + K_d(e_{[n]} - e_{[n-1]})]$$

$$G_{PD}(Z) = \frac{K_p}{1+K_d}\frac{Z-\frac{1}{1+K_d}}{Z}$$

Figure 5 shows the effect of placing the zero close to the origin (left plot) or close to 1 (right plot).

The use of the PD controller improves the system stability and dynamic response. By placing the controller zero close to one, we have a fast system response. Having the controller zero close to the origin, may lead to overshoot and ringing in the system response.

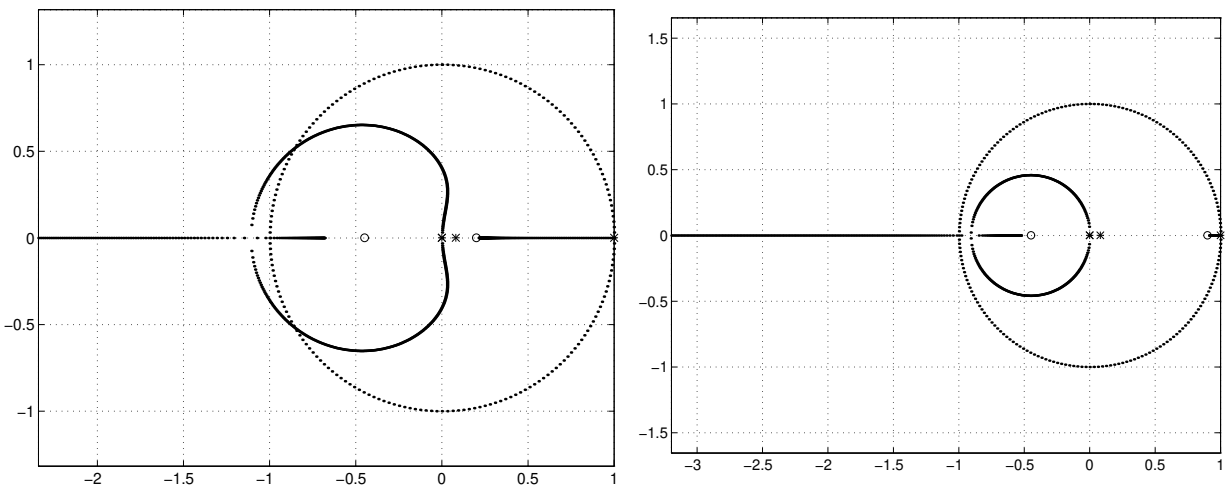

Fig. 5.: System Root-Locus using a PD controller. Left: the zero is placed close to the origin at 0.2. Right: the zero is placed close to one, at 0.9.

5 Results

The system described in the previous sections was implemented and tested. The control strategy consists in regulating to zero the angular error between the robot heading and the direction of the final point relative to the current position of the robot. Figure 6 shows the cellular robot developed for the experiments.

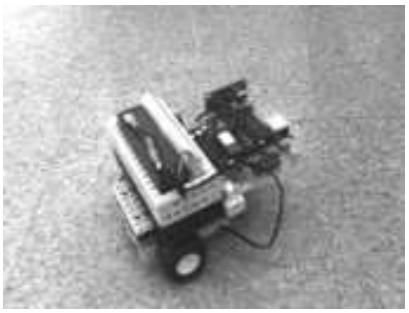

Fig. 6.: Cellular robot developed for the experimental tests.

For the tests, the projective transformation between the image plane and ground plane was estimated using four known points in the ground plane. During normal operation, the user specifies the desired end point in the image coordinates and the control system generates the suitable commands to drive the robot to this position. Figure 7 shows the image in the user's interface where the robot trajectory was superimposed. In the same figure we also show the temporal evolution of the robot heading direction when a step input is applied.

In Figure 8 we show the trajectory described by a vehicle during a maneuver together with the estimates of the vehicle position and linear velocity obtained by the $\alpha - \beta$ filter. To analyze the control system performace the figure shows also the temporal evolution of the angular error and distance to the target point.

The following experiment was done starting the robot with a large initial direction error relative to the final point. Additionally, we have manually changed the camera position, thus illustrating the robustness of the overall system to the camera position/orientation. Figure 9 shows the vehicle trajectory, estimated position and velocity, angular error and distance to the target. The robot describes a curvilinear path, moving away from the goal point and finally moving towards the desired position. The shape of the path is mainly

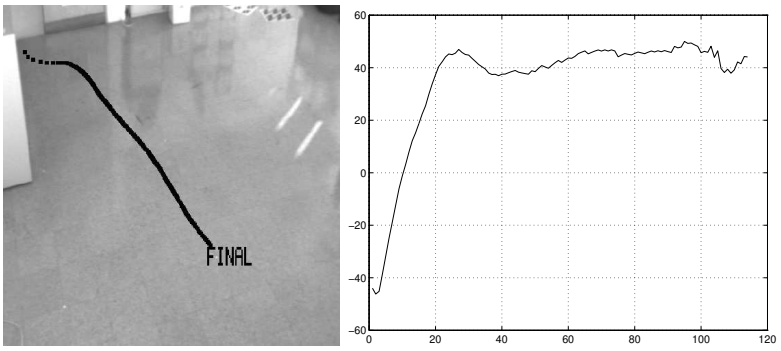

Fig. 7.: Left: Example of robot trajectory superimposed on the image. Right: Step response of the heading closed loop control system.

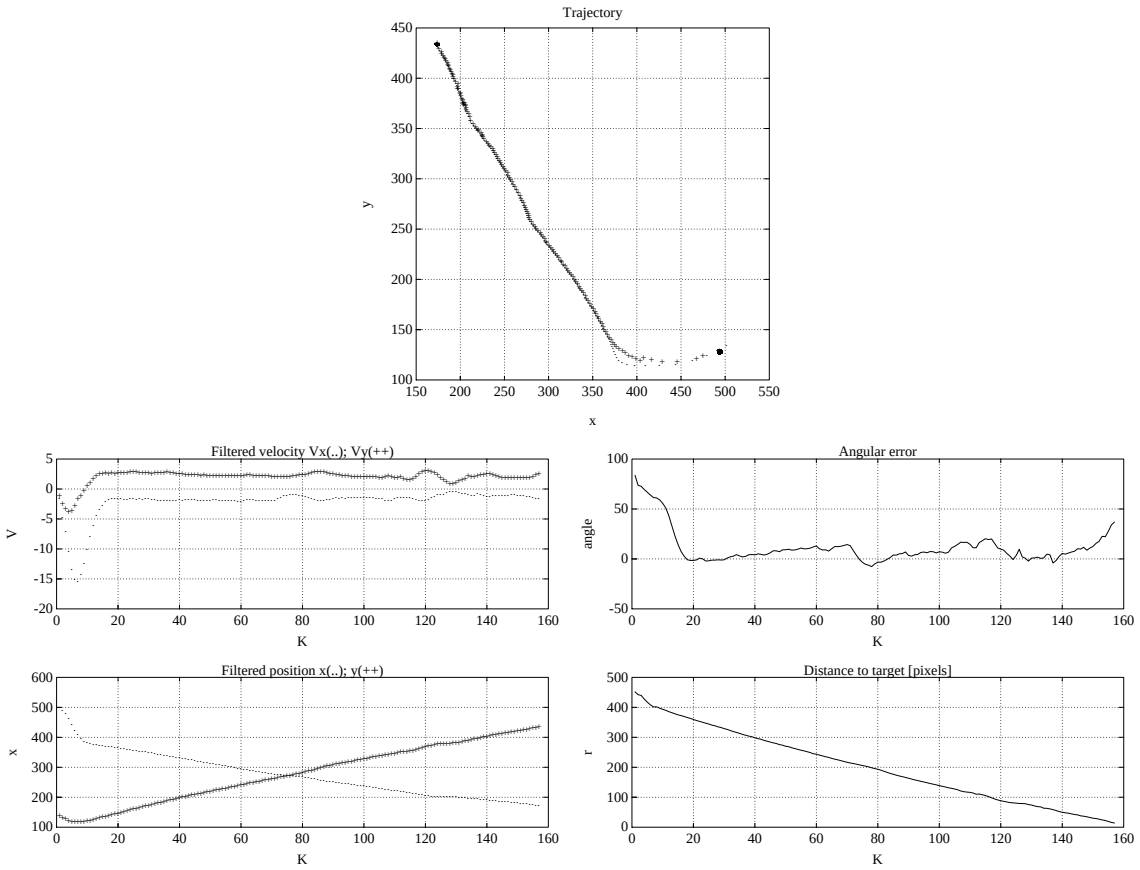

Fig. 8.: Robot trajectory during a maneuver, estimated velocities, angular error to trget and distance to goal, over time.

due to the non-holonomic constraints of the robot platform and on the saturation imposed on the control values.

Another aspect worth discussing is the inverse projection of the orientation error measured in the image back to the ground plane, using the inverse of the projection matrix P_p. If the angular errors were back projected onto the horizontal plane, then in the areas far from the camera (where the perspective effects are stonger), the measurement error noisewould be quite amplified. Therefore, in our system, the controller uses the error measured in the image plane which is directly connected to the information available to the user in the system's interface. With this approach, we achieve much smoother robot trajectories, as opposed to the case where the error signal is back projected to the ground plane. The difference between image and 3D

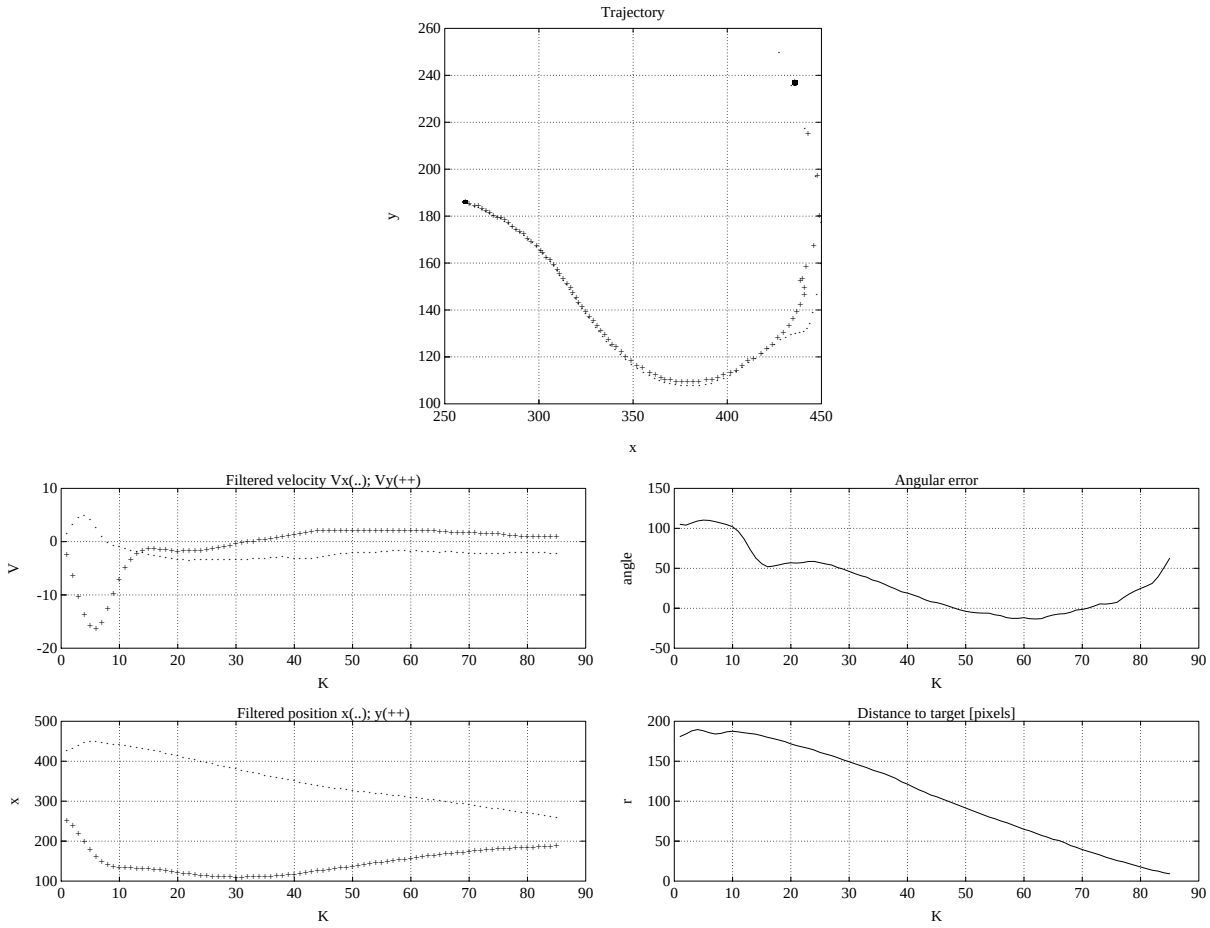

Fig. 9.: Robot trajectory during another maneuver with a large initial orientation error. The plots show evolution of the estimated velocities, angular error, and distance to goal.

position based control is discussed in [5] in the general context of visual servoing.

6 Conclusions

We have described an approach for remote control of cellular robots using visual feedback obtained from a static camera viewing the robot workspace. This system offers a flexible framework for robotic applications such as surveillance and inspection, where the user can specify goal-points in image coordinates to be reached by the robot.

This problem involves a number of aspects related to vision and control that can be carefully modeled and considered when designing the complete system. We have detailed the transformation relating the image plane and the coordinate system attached to the pavement where the robot is moving. We have also described a simple fast tracking algorithm to estimate of the vehicle position and orientation on the image plane, based on an $\alpha - \beta$ filter. Then, the various dynamic components of the complete system were modeled and we presented a suitable dynamic controller based on linear control theory. By specifying the control goal in the image coordinates, the overall accuracy of the system is independent of the errors in the calibration of the camera intrinsic or extrinsic parameters. Nevertheless, the procedure of camera calibration is analized in the paper for the cases where 3D positioning is required. Even in the case where all the data are specified in image coordinates the transformation from the image plane to the ground plane can provide useful insight as to the sign of the control signals and suitable values of gain.

As a final word, we would like to stress that such a system can be very useful in various application domains like surveillance and inspection, and may serve as an inexpensive and yet flexible testbed for

cooperation among various vehicles. Additionally, we consider that it has a large poteltial as an educational platform due to its simplicity and the large number of involved that can be carefully addressed, in vision, robotics, filtering, control, etc.

Acknowledgements

This work was partially funded by projects JNICT-PBIC/TPR/2550/95 and PRAXIS/3/3.1/TPR/23/94. The authors wish to thank Mr. Nuno Pereira for his help in developing the robot hardware and Prof. Vitor Barroso for the stimulating discussions and valuable advice for the design of the tracking filtering procedure.

References

1. M. Eklund, G. Ravichandran, M. Trivedi, and S. Marapane. Real-time visual tracking using correlation techniques. In *Proc. of the 2nd. IEEE Workshop o Applcations of Computer Vision*, Florida, USA, December 1994.
2. B. Espiau, F. Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.
3. O. Faugeras. *Three Dimentional Computer Vision*. MIT Press, 1996.
4. A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.
5. S. Hutchinson, G. Hager, and P.I. Coorke. A tutorial in visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.
6. P. Kalata. The tracking index: a generalized parameter for $\alpha-\beta$ and $\alpha-\beta-\gamma$ target trackers. *IEEE Transactions on Aerospace and Electronic Systems*, 20(2):174–182, March 1984.
7. K. Ogata. *Discrete-Time Control Systems*. Prentice-Hall, 1995.
8. C. Zanardi, J.-Y.-Hervé, and P. Cohen. Mutual learning of unsupervised interaction between mobile robots. In *Proc. 13th. IEEE Int. Conf. Pattern Recognition - ICPR*, Vienna, Austria, Aug. 1996.

This article was processed using the TEX macro package with SIRS97 style