To appear in *Robotics and Autonomous Systems*, 1998

Vision Based Remote Control of Cellular Robots

José Santos-Victor

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

Abstract

This paper describes the development and design of a vision-based remote controlled cellular robot. Cellular robots have numerous applications 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 monitor the working area. As a result, a robust control of the robot trajectory is achieved without depending on the camera calibration. The remote user simply specifies 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, including the stability analysis when the camera location is unknown. 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 many applications (surveillance, inspection, mine localization, space exploration, underwater, etc). We address this problem by designing simple and inexpensive cellular robots that can individually or cooperatively operate in a large remote area [14].

Although we focus mainly on the design of a single agent, cost is an important constraint, if multiple robots are to be used. 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 robot performs various tasks. The user specifies commands simply by indicating in the image, the desired end position for each robot. The robot is controlled through visual feedback

[5], and 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 is an important remark since it simplifies the installation and system operation by not requiring the calibration of the camera coordinate system relative to that of the workspace. This general setup is illustrated in Figure 1.

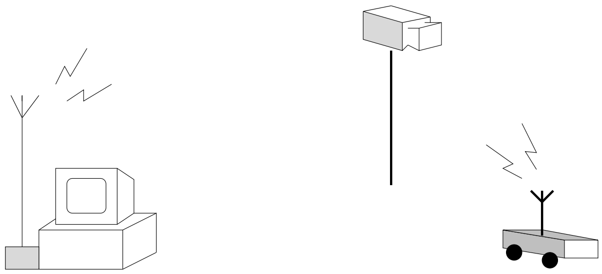

Figure 1: Overall system setup

We can extend this paradigm to the case where a robot is capable of several local autonomous behaviors that may be launched or coordinated by a remote user. In this way, the user is not subject to long, tedious, routine operations which can be performed by autonomous behaviors [12]. On the other hand, with this supervised autonomy mode, the whole system [2, 10] can benefit from the presence of an operator in the loop, in terms of security and performance.

Currently, the control strategy is based on the heading direction. We control the vehicle angular velocity such that it moves in the direction of the goal. The linear velocity is kept constant during the entire path, but could also be controlled as a function of the distance to the final point.

Another aspect worth mentioning is related to the camera (sensor) and ground plane (world) coordinate systems. The overall goal is to control the vehicle heading direction towards a given scene point. However, all the measurements (e.g vehicle position and orientation) are taken on the image plane, thus consisting of projections of the vehicle real position and orientation.

One solution is to determine the relationship between image measurements and the real variables and always transform these measurements into the vehicle, or ground plane coordinate system. The drawback is that innacuracies in the image measurements would be projected onto the ground plane in a non uniform manner, as each image pixel subtends a different area of the ground plane. The alternative solution is to control the vehicle based on the image measurements directly, without explicitly performing coordinate transformations. Then, the system performance is no longer dependent on the calibration parameters, and the control problem is defined on the sensor coordinate frame directly. In our system, we can use both control alternatives, but direct image based control is the preferred choice.

The system we present consists of a single robot, controlled via a radio link from a PC. The host computer executes the tasks of determining the robot position and orientation in the image coordinates, and generates the proper control signals. The visual tracking system runs at about 8Hz, without using any special image processing hardware. The communication is assured via serial port, with a simple protocol to send the control commands.

Currently, the robot is assembled from LEGO components and contains a CPU-board consisting of the radio receiver, decoding logic and PWM (pulse width modulation) generation for the differential control of the robot. Each robot is identified by a 4 bit address and hence, we can control up to 16 cellular vehicles. There are no on-board sensors. See [1] for more details.

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.

In the paper we describe thoroughly the problem of designing a digital control for the system, and a robust tracking procedure. Additionally we show that the system is (locally) stable independently of the unknown camera position and orientation.

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

The system described in this paper involves two coordinate frames. The *world* frame is used to describe the robot position on the ground plane. The *camera* coordinate system is required to express the robot position on the image plane.

One way to develop the control system for the robot trajectory, is to convert all the image measurements back to the world coordinate frame. An alternative approach consists in using the image measurements directly to control the motion, without an intermediate step of coordinate transformation. This latter approach avoids the need for estimating the projective transformation between the image and ground planes.

For the sake of completeness, we fully describe the calibration procedure in Appendix A, if the reader prefers to use the calibrated version. The first problem now is to determine the vehicle position and velocity in the image.

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

To control the robot motion, we must sense its position and heading direction over time, using the camera monitoring the work space. A strong requirement is to obtain these measurements at a frequency allowing the closed loop control strategy [4].

Since the camera is fixed, the vehicle image position is estimated simply 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 wider 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 instantaneous velocity.

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

$$X(k) = [x \quad \dot{x}]^{\mathrm{T}} \qquad\qquad Z(k) = x \tag{1}$$

The dynamic state-space model, assuming constant velocity, is given by:

$$\begin{aligned} X(k+1) &= \phi X(k) + \psi w(k) \\ \\ Z(k) &= C^T X(k) + \eta(k) \\ &= [1 \quad 0] X(k) + \eta(k) \end{aligned} \tag{2}$$

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:

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

where T denotes the sampling period.

Notice that, as the vehicle is under closed loop control, the target maneuvers may be known, at least partially. The reason to consider the target maneuver as an unmodeled perturbation, is that there are various unknown parameters and non linearities in the real robot, that would need to be identified for a more accurate system model. Such a model would allow the use of an extended Kalman-Bucy filter for the state estimation.

The problem then, is to estimate the state vector using the set of observations available up to the current sampling instant:

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

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

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

[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 the motion model along the x and y directions is decoupled, we can apply the $\alpha - \beta$ filter independently in both directions.

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:

$$P(k|k) = P(\mathcal{F}, \sigma_\eta^2, \sigma_w^2).$$

Since the system is linear and the noise processes are white, the optimal mean-squared-error (MSE) filtering is the Kalman filter [7]. We can now use the *tracking index* introduced in [9] that characterizes the general tracking problem and further simplifies the optimal tracking solution.

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

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

The tracking index completely describes the tracking solution since once λ is determined, the optimal filter is implicitly defined, together with the tracking performance criterion:

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

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:

$$K = [\alpha \qquad \beta/T]^T$$

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 bandwidth.

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.

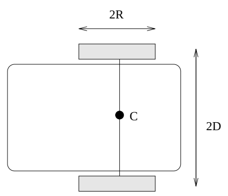

Figure 2: Vehicle Kinematics

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 half-distance between the wheels, D, and the radius of the wheels R, according to :

$$\begin{bmatrix} V_c \\ \omega_c \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ -R/D & R/D \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{3}$$

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 velocity and the input voltage is well approximated by a first-order dynamic system[11] :

$$\omega(s) = \frac{K_\omega}{\tau s + 1} U(s) \tag{4}$$

where τ is the mechanical time constant, and K_ω is the steady state gain.

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 velocities of the vehicle reference point:

$$\begin{aligned} \omega_C(s) &= \frac{R}{D}\frac{K_\omega}{\tau s + 1} U_{diff}(s), \qquad & U_{diff}(s) = U_R(s) - U_L(s) \\ V_C(s) &= \frac{R}{2}\frac{K_\omega}{\tau s + 1} U_{comm}(s), \qquad & U_{comm}(s) = U_R(s) + U_L(s) \end{aligned} \tag{5}$$

In conclusion, by using 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

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. In addition, 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 [11] 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 is shown in Figure 3, including an impulsive sampling block, and a zero-order-hold circuit at the controller output. The

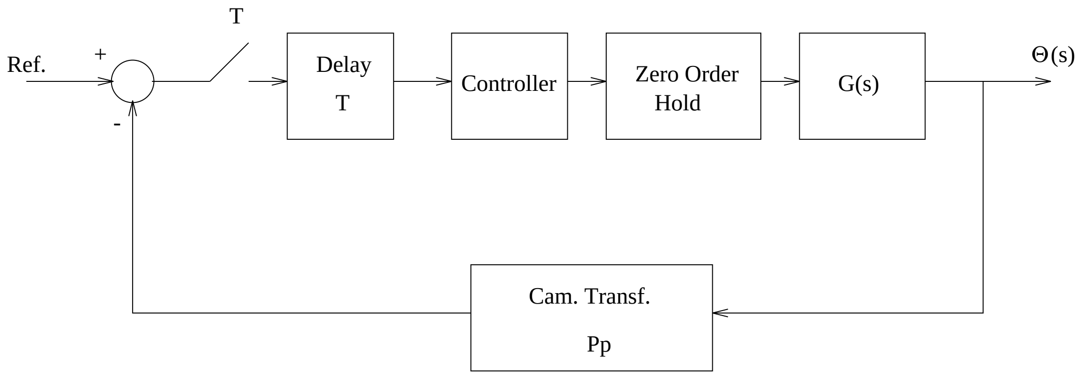

Figure 3: Control system

transfer function $G(s)$ relates the vehicle heading to the differential voltage U_{diff}:

$$G(s) = \frac{R}{D}\frac{K_\omega}{s(\tau s + 1)} \tag{6}$$

The impulsive transfer function including the Zero Order Hold circuit and the process $G(s)$ can be computed from the Z-transform of the sampled input and output signals:

$$\begin{aligned} G_{eq}(Z) &= \mathcal{TZ}\left\{\frac{1-e^{-Ts}}{s}\cdot\frac{R}{D}\frac{K_\omega}{s(\tau s+1)}\right\} \\ &= (1-Z^{-1})\frac{R}{D}\mathcal{TZ}\left\{\frac{K_\omega}{s^2(\tau s+1)}\right\} \\ &= \frac{RK_\omega\tau}{D}\frac{AZ+B}{(Z-1)(Z-e^{-T/\tau})} \end{aligned} \tag{7}$$

where we have defined

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

One has still to consider the delay of one sampling period, introduced by the image processing. In the frequency domain, the delay is expressed by a factor Z^{-1}. Thus, the discrete open loop transfer function is the following:

$$G_{OL}(Z) = \frac{RK_\omega\tau}{D}\frac{AZ+B}{Z(Z-1)(Z-e^{-T/\tau})} \tag{8}$$

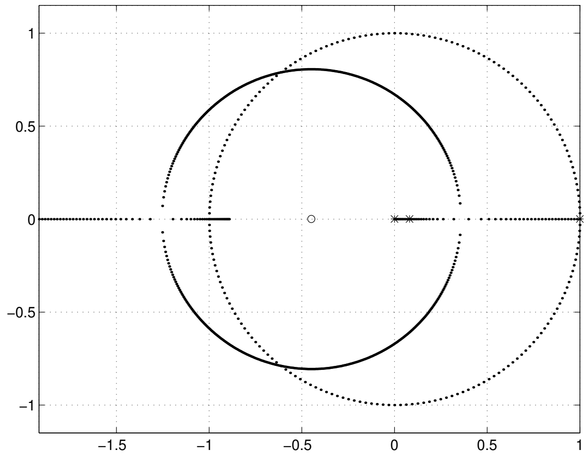

Figure 4: System Root-Locus when a proportional controller is used.

The root-locus corresponding to this open-loop discrete-time transfer function is shown in Figure 4 [2]. Hence, for a certain range of the gain, the system is stable and becomes unstable for large gains. An 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:

$$
\begin{aligned}
u_{PD}[n] &= K_p[e_{[n]} + K_d(e_{[n]} - e_{[n-1]})] \\
G_{PD}(Z) &= K_p(1+K_d)\frac{Z - \frac{K_d}{1+K_d}}{Z}
\end{aligned}
$$

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.

[2] We have used $T = 0.125s$ and the system pole at $2Hz$, yielding $\tau = 0.08$. The open loop poles are located at $\{0, 1, 0.08\}$ and the zero at $\{-0.4487\}$.

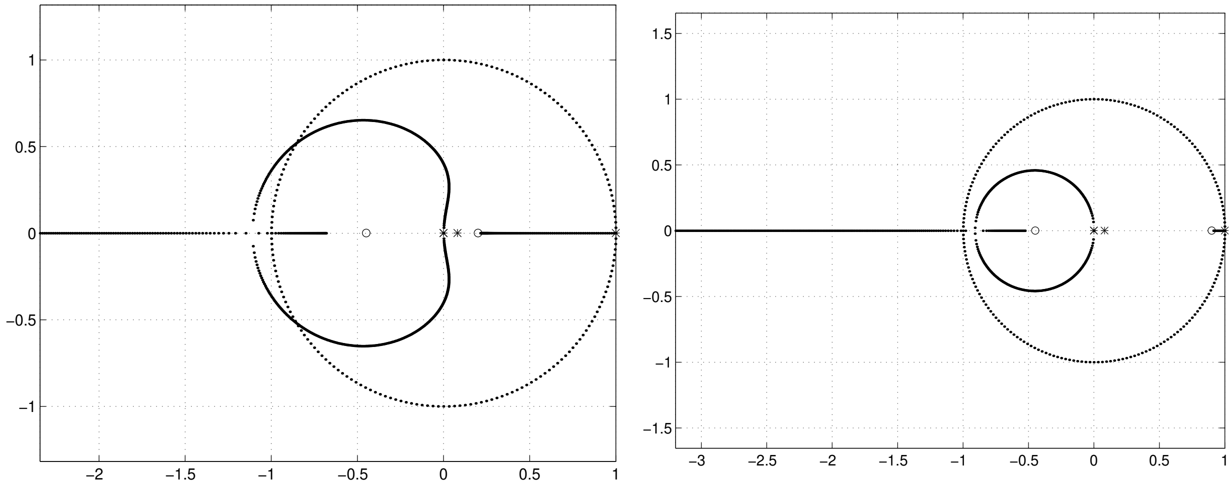

Figure 5: System Root-Locus using a PD controller. Left: the zero is placed close to the origin at 0.2. Right: the zero is placed close to one, at 0.9.

4.1 Stability Analysis

We have discussed the control system design problem considering that the measurements of the vehicle heading direction could be directly obtained. However, we only have access to *projections* of these measurements, on the image plane. The question that arises then is whether or not, the transformation introduced by the camera observations will alter the system stability.

Assuming that such transformation is known (for the calibration procedure refer to Appendix A), one could always convert image coordinates to ground plane coordinates. The challenging problem however, is to design a stable controller, assuming the camera calibration is unknown. We show that the system is (locally) stable independently of the (unknown) camera position and orientation, using Liapunov's second method for continuous systems [3].

We will use a continuous-time second order system to describe the relationship between the torque, μ, applied at the vehicle mid-point and the heading direction, θ (see Section 3.2) :

$$M\ddot{\theta} + C\dot{\theta} = \mu \tag{9}$$

where M describes the system mass and C is a positive constant related to the mechanical pole. The control system error, θ_e, is defined as the difference between the vehicle heading and some specified reference direction, θ^r:

$$\theta_e = \theta - \theta^r \tag{10}$$

The mapping between the angles, θ, measured on the ground plane and the corresponding angles projected on the image plane, θ_I, is denoted by a function $\mathcal{H}$:

$$\theta = \mathcal{H}(\theta_I) \tag{11}$$

This function describes the viewing geometry, i.e. the correspondence between an angle measured in the image plane and the corresponding angle measured on the ground floor. In general, this function depends on the orientation of the image plane relative to the ground plane, on the viewing direction, on the angle being measured and, finally on the absolute angle orientation.

The mapping $\mathcal{H}(\theta_I)$ is differentiable, provided that the camera plane is not perpendicular to the ground plane, that would give rise to singularities. Using equation (9) and the differentiable (projective) mapping (11), we can describe the dynamic evolution of the vehicle heading, as seen in the image plane:

$$M\mathcal{H}'(\theta_I)\ddot{\theta}_I + M\mathcal{H}''(\theta_I)\dot{\theta}_I^{\,2} + C\mathcal{H}'(\theta_I)\dot{\theta}_I = \mu \tag{12}$$

where we have used

$$\mathcal{H}'(\theta_I) = \frac{\partial \mathcal{H}(\theta_I)}{\partial \theta_I} \qquad \mathcal{H}''(\theta_I) = \frac{\partial^2 \mathcal{H}(\theta_I)}{\partial \theta_I^2} \tag{13}$$

If we assume that $\mathcal{H}(\theta_I)$ varies slowly, we can use a linearized model to describe the image evolution of the vehicle heading:

$$M\mathcal{H}'\ddot{\theta}_I + C\mathcal{H}'\dot{\theta}_I = \mu \tag{14}$$

where the constant $\mathcal{H}'$ is the slope of $\mathcal{H}(\theta_I)$ evaluated at a given angular position. One can now build a Liapunov function combining the system kinetic energy and a quadratic term in the orientation error :

$$V(\theta_I, \dot{\theta}_I) = \frac{1}{2} M\mathcal{H}'\dot{\theta}_I^{\,2} + \frac{1}{2} K_p(\theta_I^r - \theta_I)^2 \tag{15}$$

where K_p is a positive constant, and we assume for the moment that $\mathcal{H}'$ is positive. In these conditions, $V(\theta_I, \dot{\theta}_I)$ vanishes for the desired equilibrium point, $(\theta_I, \dot{\theta}_I) = (\theta_I^r, 0)$, and is positive elsewhere. Computing the time derivative of $V(\theta_I, \dot{\theta}_I)$ yields:

$$\begin{aligned}
\dot{V}(\theta_I, \dot{\theta}_I) &= M\mathcal{H}'\dot{\theta}_I\ddot{\theta}_I - K_p(\theta_I^r - \theta_I)\dot{\theta}_I \\
&= \dot{\theta}_I\{M\mathcal{H}'\ddot{\theta}_I - K_p(\theta_I^r - \theta_I)\} \\
&= \dot{\theta}_I\{\mu - C\mathcal{H}'\dot{\theta}_I - K_p(\theta_I^r - \theta_I)\}
\end{aligned}$$

If we choose a control law given by:

$$\mu - K_p(\theta_I^r - \theta_I) = -K_v\dot{\theta}_I \tag{16}$$

with K_v being a positive scalar, it yields :

$$\dot{V}(\theta_I, \dot{\theta}_I) = -\{C\mathcal{H}' + K_v\}\dot{\theta}_I^{\,2} \tag{17}$$

Since C and K_v are positive scalars, and $\mathcal{H}'$ was also assumed to be positive, $\dot{V}(\theta_I, \dot{\theta}_I)$ is a negative semi definite function. Hence $(\theta_I, \dot{\theta}_I) = (\theta_I^r, 0)$, is a stable equilibrium point.

As the system described by equation (12) is an autonomous system, to study the asymptotic stability we need to invoke LaSalle's theorem [13], considering the region where $\dot{V}(\theta_I, \dot{\theta}_I)$ vanishes:

$$\begin{aligned}
\Gamma &= \left\{(\theta_I, \dot{\theta}_I) \in \Re^2 : \dot{V}(\theta_I, \dot{\theta}_I) = 0\right\} \\
&= \left\{\theta_I \in \Re, \dot{\theta}_I \in \Re = 0\right\}
\end{aligned}$$

Among all the points $(\theta_I, \dot{\theta}_I = 0)$, only the desired equilibrium point $(\theta_I, \dot{\theta}_I) = (\theta_I^r, 0)$ is a solution of the system equation (14). We can conclude that the control system is asymptotically stable, and the original non-linear system is *locally* asymptotically stable. The control law is then given by:

$$\mu = K_p(\theta_I^r - \theta_I) - K_v\dot{\theta}_I \tag{18}$$

which is a PD controller based on image measurements alone that ensures (local) stability of the closed-loop system.

The mapping $\mathcal{H}$ depends both on the camera position and orientation relative to the ground plane, and on the particular image pixel and angle under consideration. We have assumed that $\mathcal{H}'$ is always positive. This hypothesis means that for an arbitrary camera location, an increase of the angle between two image lines, h_1 and h_2, is due to an increase of the angle between by the corresponding lines, H_1 and H_2, on the ground plane. In other words, when the angle increases on the image plane, the same happens with the angle generated by the corresponding lines on the pavement. This idea is illustrated in Figure 6. This hypothesis holds for any camera orientation, except when the image plane becomes

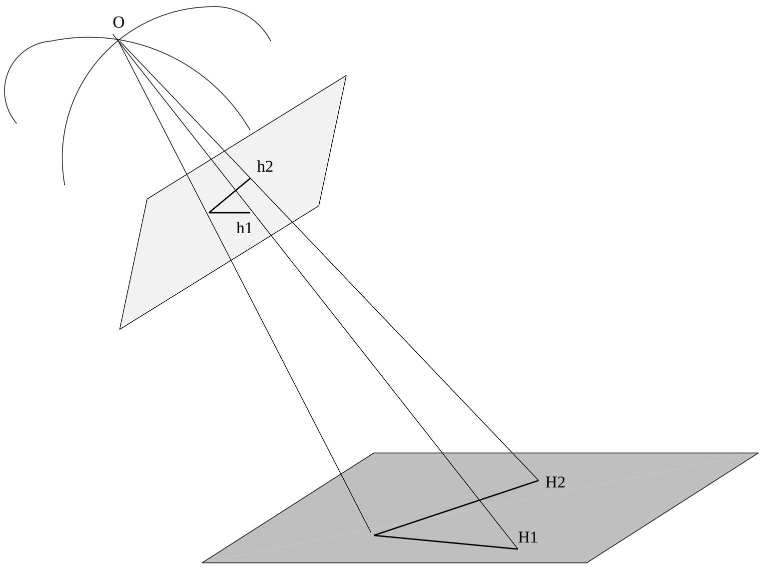

Figure 6: Increasing the angle between two image lines, h_1 and h_2, yields an increase of the angle between the corresponding lines, H_1 and H_2, drawn on the ground plane.

perpendicular to the ground plane, or the viewing direction tends to become perpendicular to the optical axis.

If the image plane is parallel to the ground plane, then $\mathcal{H}'$ is constant (equal to one) all over the image plane; the model linearization is no longer an approximation, and the closed loop control system is *globally* asymptotically stable.

The analysis of the global stability of the non-linear system (12) can be made in a similar way. However, it requires the knowledge about upper and lower bounds of $\mathcal{H}'$ to implement the resulting control law.

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 7 shows the cellular robot developed for the experiments.

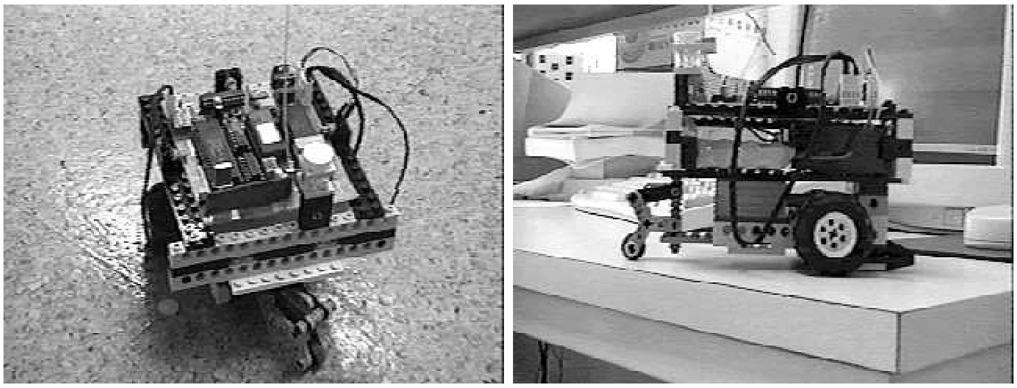

Figure 7: Cellular robot developed for the experimental tests.

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 8 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.

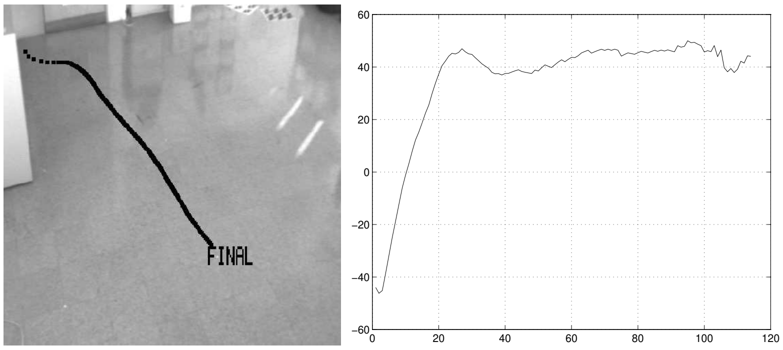

Figure 8: Left: Example of robot trajectory superimposed on the image. Right: Step response of the heading closed loop control system.

In Figure 9 we show the trajectory described by the 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 performance the figure shows also the temporal evolution of the angular error and distance to the target point. The position and velocity estimates obtained with the $\alpha - \beta$ filter are smooth and provide reliable data to determine the heading direction. Notice also that the angular error increases when approaching the final goal, when small misalignements yield large angular errors. The last plot shows the distance to the target measured *perpendicularly* to the heading direction. As expected, the distance to goal decreases linearly, as the linear velocity is kept constant.

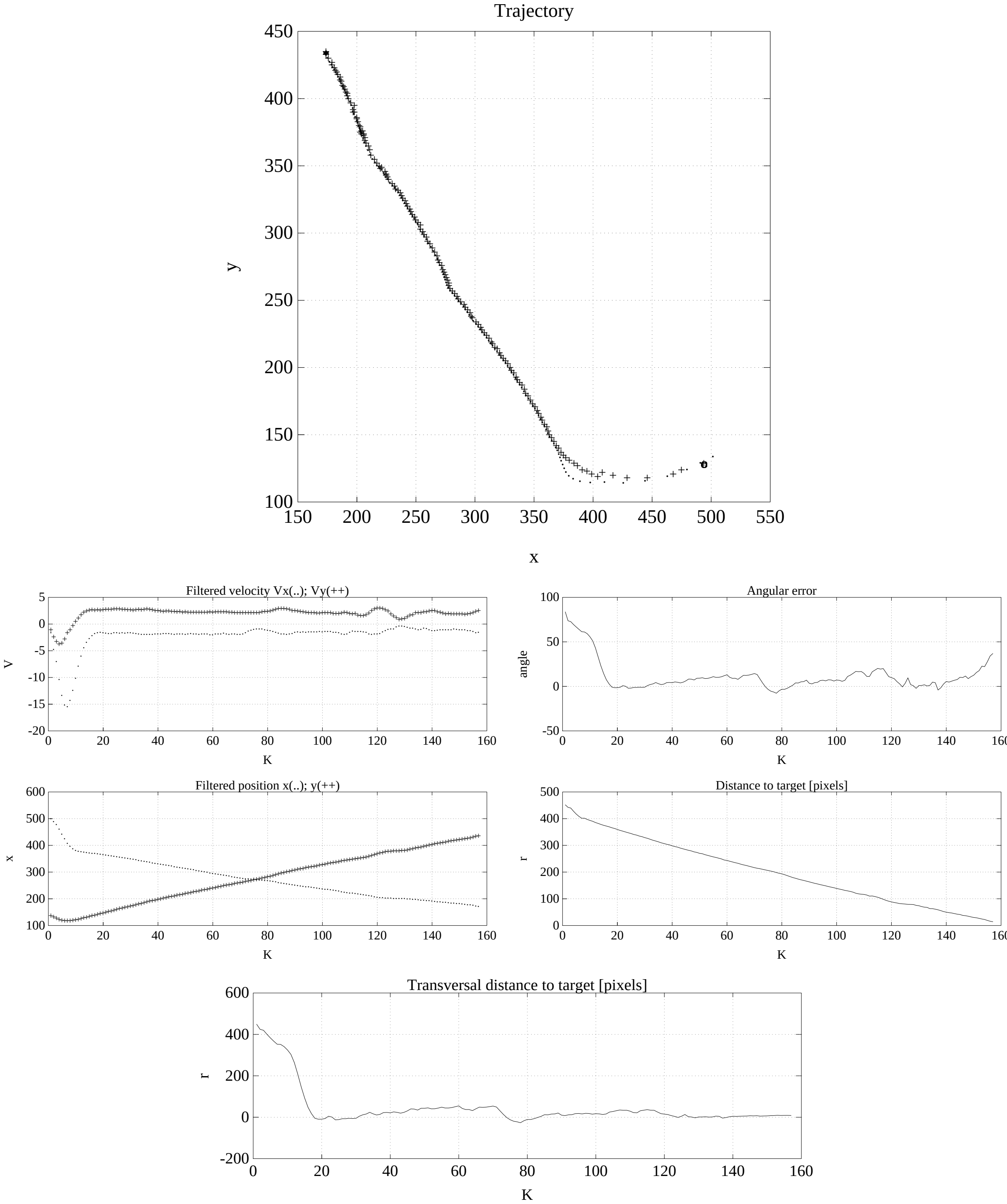

Figure 9: Robot trajectory during a maneuver, estimated velocities, angular error to target and distance to goal, over time. The angular error tends to increase during the final approach, since at short distances a small misalignement corresponds to a large angular deviation. However, the distance to the target measured perpendicularly to the robot heading (last plot) approaches zero in a consistent manner.

In the final experiment, the robot was initialized with a large orientation error relative to the final point. Additionally, we have manually changed the camera position, to illustrate the robustness of the overall system to the camera position/orientation. Figure 10 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, towards the desired position. The shape of the path is mainly 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 (see Appendix). If the angular errors were back projected onto the horizontal plane, then in the areas far from the camera (where the perspective effects are stronger), the measurement error noise would 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 position based control is discussed in [8] 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. The details about the transformation relating the image plane and the coordinate system attached to the pavement, where the robot is moving, is given in Appendix A. 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

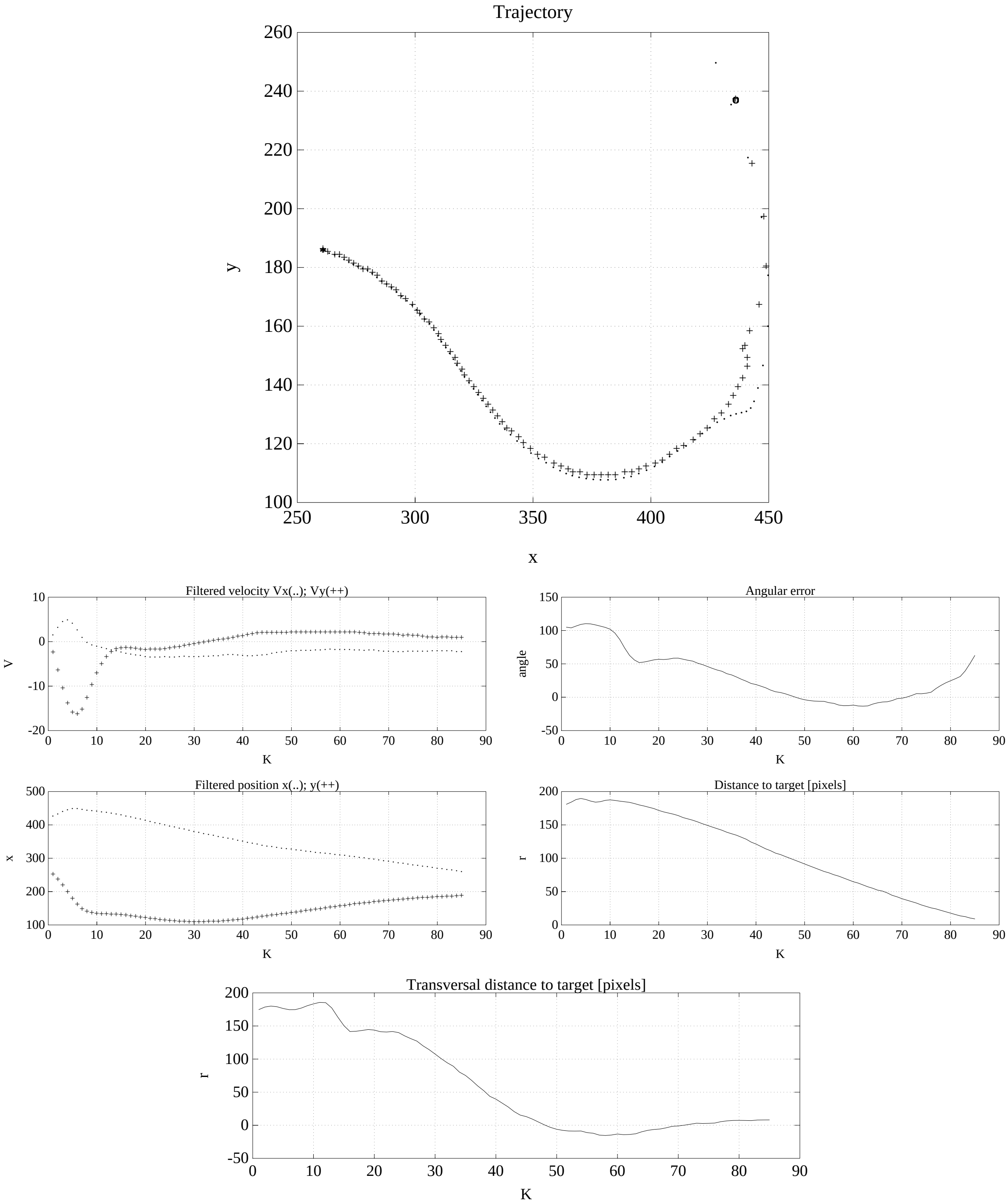

Figure 10: Robot trajectory during another maneuver with a large initial orientation error. The plots show the evolution of the estimated velocities, angular error, and distance to goal. The final increase of the angular error is due to the effect already explained in Figure 9. The bottom plot shows how the distance to the target, measured perpendicularly to the heading direction, tends to zero with a small overshoot.

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 analyzed in the appendix for the cases where 3D positioning is required.

We have also shown that the closed-loop control system is (locally) stable for any camera position/orientation except when the image plane becomes perpendicular to the ground plane. Naturally, the performance will be different when the camera position is changed. The problem of addressing the optimal control parameters for *any* camera position orientation is being considered for future work.

One aspect worth mentioning is that the proposed scheme can work in situations where the ground is not necessarily planar. The main constraint then is related to some degree of smoothness of the ground surface to preclude self-occluding concavities. Currently a new full-terrain robot is under construction for outdoor applications in uneven terrain.

Another interesting extension is the use of multiple cameras observing a large environment. From a theoretical point of view this is exactly the same problem as the one described in this paper. If the projection matrices are estimated, then we can relate pixels of the various images, simply by composing the various projective homographies.

Alternatively, if we prefer to keep the system uncalibrated we have shown that the control system does not depend on the information about the cameras geometry. In this case, it would not be necessary to explicitly relate the coordinate systems of the various images, as the control could be easily switched between the various cameras. Alternatively, if both the robot and the goal are visible in more than one camera, one could use the multiple angular errors in the global robot controller, thus explicitely benefitting from the existence of multiple measurements to further improve the accuracy.

We have discussed that although stability is independent of the camera geometry, thus enlarging the range of applicability of the system, performance is not. To overcome this difficulty, one could locally use the robot motion to adapt the controller parameters in order to obtain a uniform performance over the entire image and across various viewing geometries.

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 potential as an educational platform due to its simplicity and the large number of involved problems that can be carefully addressed, in vision, robotics, filtering, control, etc.

A 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 [6].

$$\begin{aligned} \tilde{m} &= \tilde{P}\tilde{M} \\ \begin{bmatrix} \lambda x_p \\ \lambda y_p \\ \lambda \end{bmatrix} &= \begin{bmatrix} fk_u & 0 & u_o & 0 \\ 0 & fk_v & v_o & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \end{aligned} \tag{19}$$

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 projection matrix, that depends on the camera intrinsic parameters (k_u, k_v, u_o, v_o, f) and the camera position relative to the world frame (r_{ij}, t_x, t_y, t_z).

As the camera is observing a plane where the vehicle moves, 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 uniquely to a certain point on the plane, according to :

$$\begin{aligned} \tilde{m} &= \tilde{P}_p\tilde{M} \\ \begin{bmatrix} \lambda x \\ \lambda y \\ \lambda \end{bmatrix} &= \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \end{aligned} \tag{20}$$

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

Equation (20) 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 a minimum of 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 [6].

Acknowledgements

This work was partially funded by research projects JNICT-PBIC/TPR/2550/95 and PRAXIS/3/3.1/TPR/23/94. The authors wish to thank Prof. Vitor Barroso for the stimulating discussions and valuable advice for the design of the tracking filtering procedure. Carlos Carreira was responsible for the construction of the robot that was used for this work.

References

[1] C. Carreira and J. Santos-Victor. Vision based teleoperated cellular vehicles. In *Proc. of 5th Int. Symposium on Intelligent Robotic Systems*, Stockholm, Sweden, July 1997.

[2] G. Cheng and A. Zelinski. Supervised autonomy: A paradigm for teleoperating mobile robots. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, September 1997.

[3] P.M. Derusso, R.J. Roy, and C.M. Close. *State Variables for Engineers*. John Wiley and Sons, 1965.

[4] M. Eklund, G. Ravichandran, M. Trivedi, and S. Marapane. Real-time visual tracking using correlation techniques. In *Proc. of the 2nd. IEEE Workshop on Applications of Computer Vision*, Florida, USA, December 1994.

[5] 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.

[6] O. Faugeras. *Three Dimentional Computer Vision*. MIT Press, 1996.

[7] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

[8] 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.

[9] P. Kalata. The tracking index: a generalized parameter for $\alpha - \beta$ and $\alpha - \beta - \gamma$ target trackers. *IEEE Trans. Aerospace and Electronic Systems*, 20(2):174–182, March 1984.

[10] R.C. Luo and T.M. Chen. Remote supervisory control of a sensor based mobile robot via internet. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Grenoble, France, September 1997.

[11] K. Ogata. *Discrete-Time Control Systems*. Prentice-Hall, 1995.

[12] J. Santos-Victor and G. Sandini. Embedded visual behaviors for navigation. *Robotics and Autonomous Systems*, 19(3-4):299–313, March 1997.

[13] J.L. Willems. *Stability Theory of Dynamical Systems*. Thomas Nelson & Sons, 1970.

[14] C. Zanardi, J.-Y.-Hervé, and P. Cohen. Mutual learning of unsupervised interaction between mobile robots. In *Int. Conf. Pattern Recognition - ICPR*, Vienna, 1996.