

MODELLING AND CONTROL OF A VISUAL SERVOING SYSTEM

M. VARGAS*, A. R. MALPESA and F. R. RUBIO

*Dept. Ingeniería de Sistemas y Automática,
Escuela Superior de Ingenieros, Universidad de Sevilla,
Camino de los Descubrimientos s/n, 41092-Sevilla (Spain)*

(Received 1 February 1999)

In the present paper, the modelling and control of a visual servoing system is presented. The main different approaches to face this visual servoing problem are presented. Among them, the position-based approach is chosen. In this approach, the features extracted from the images are used to estimate the relative object-camera position. This position is then compared with a desired relative position (used as a reference for the servoing system). The preliminary calibration procedures needed for this approach are also outlined in the paper.

As a first approximation to our visual servoing implementation, the planar case is considered in simulation and real experiments. In the general case, the six degrees of freedom (DOF), necessary to reach any position and orientation inside the robot working space, are involved. The extension to this general case is also presented, but no experimental results have been obtained yet.

The robot manipulator used in our implementation is the PUMA 560. Unlike other authors, the control commands given to this robot are not exercised directly on the joint's servomotors. Instead, the robot is commanded at a higher level, using the *ALTER* facility, in which the references to the robot are given in cartesian coordinates, avoiding the need of inverse kinematics calculations.

The portion of the model that takes in the behaviour of the robotic system will be considered as a low-level control stage. Its sampling period is fixed by the robotic system at 28 ms. On the other hand, the visual control stage, or higher-level stage, comprises the camera model, the image capturing mechanism, the position estimation algorithm, *etc.* The sampling period of this second portion of the model is fixed to a value between 3 and 9 times the lower-level sampling period, depending on the complexity of the feature extraction procedure. After this description, it is obvious that our system has a multirate nature.

A detailed model for the described visual servoing system is obtained in the paper. However, in order to ease the controller design process, a simplified, single-rate model is calculated. Based on this reduced model, proportional, pole placement and Smith predictor controllers have been designed and tested by simulation and in real experiments.

*Corresponding author. Tel.: +34 954487340, Fax: +34 954487340, e-mail: vargas@cartuja.us.es

Keywords: Visual servoing system; vision perception; visual feedback; position-based approach; image-based approach; feature extraction; robot control

1. INTRODUCTION

The importance of vision perception in robotic systems is mainly due to the fact that it gives a very rich information of the environment (if it can be interpreted in a convenient way), without the need of contact or short-distance measurements. One of the drawbacks of vision sensors is that they give just a planar projection of the real world. Later processing is necessary to obtain three-dimensional information from these views.

In many applications, visual sensors are used in a static setup. In our case, the interest is on non-stationary camera placement applications. The camera will be attached to a maneuverable mechanical structure that makes possible to drive the sensor to a desired position at each instant. Among the most commonly used mechanical structures for this purpose are: pan and tilt units, robotic arms or mobile robots.

Furthermore, the visual information will be used as a feedback 'signal' in order to guide the manipulator in accomplishing its task, when it has to interact with objects in the environment. The term *visual servoing* is used in the literature to refer to this kind of setups.

The convenience of visual servoing appears when a robot (in a generic sense) has to operate or interact with moving parts or objects, whose position and orientation is unpredictable or known with some uncertainty. Usually, as it is our case, it is desired to keep a fixed relative object-camera (or object- end-effector) position, despite of the own object motion. In other cases, this desired relative position is not constant, but a function of time. As an example, consider a case in which the robot has to approach to and to pick up a part, or assembly another part to it. This later case, however, does not introduce any further difficulty in relation to the constant reference case.

Some suitable applications for visual servoing can be: manufacturing, fruit picking, teleoperation, missile tracking devices, autonomous mobile robots, *etc.* In many applications, the objects of interest may appear on a conveyor belt or similar, what means that a simple planar processing is adequate. However, from the point of view of image

processing and visual control, the complete tridimensional problem has much more interest.

This chapter describes the first approach to the implementation of visual servoing system, composed by a robot manipulator, a CCD camera attached to its end-effector (what is usually referred as *eye-in-hand configuration*), image acquisition hardware and a personal computer. The purpose of this system will be to keep the relative object-camera pose¹ invariable, reacting to the object self-motion in a convenient way. In the present work, only the planar implementation of the visual servoing system has been experimentally carried out.

The structure of the chapter is as follows: First, Section 2 presents the main types of visual servoing approaches. In Section 3, a model for the complete system is obtained and validated using just one degree of freedom (1-DOF). The necessary extensions of this model to the 6-DOF case are also presented. In Section 4, the simplified model, suitable for control design, is calculated. The final part of this section is concerned with the design and experimental tests of several controllers. Finally, the chapter concludes with a brief description of the components used in the implementation, in Section 5, and a summary in Section 6.

2. SERVOING APPROACH

In the visual servoing problem, some distinguishing features of the object of interest are extracted from the grabbed images, and used to control the pose of the robot manipulator. Depending on the way these features are introduced in the feedback loop, several approaches to the visual servoing problem can be distinguished ([1, 2]):

- *Position-based Approach* In this case, the features extracted from the images are used to estimate the position of the object, relative to the camera, and are compared with a desired relative position. Figure 1a shows a block diagram corresponding to this approach.
- *Image-based Approach* Now, the object's features are used as feedback information just as they are. The reference input is also

¹The term *pose*, or *transformation*, is used here to refer to position and orientation together, giving six coordinates, which can be also expressed by an homogeneous transformation matrix.

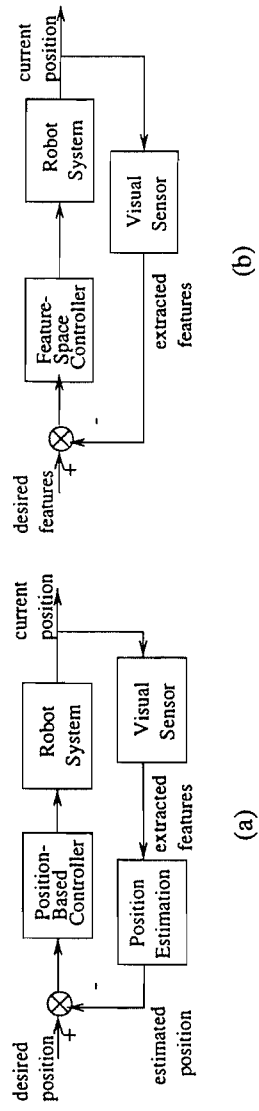


FIGURE 1 (a) Position-based configuration. (b) Feature-based configuration.

given in the feature space (see Fig. 1b). In this second case, the process of estimating the object position, starting from the observed object's features is avoided, which can save some computation time. However, the controller design is much more difficult in this case, as the image features are a complex function of the viewpoint (and, hence, of the robot position).

The approach followed in our work is the position-based approach. Despite of being simpler, this approach has the drawback of requiring some calibration procedures:

- *Intrinsic Camera Calibration* Needed to estimate the internal parameters of the camera: focal length, lens distortion, pixel aspect ratio, etc. (see, for a more detailed description, [1, 3, 4]).
- *Hand-Eye Calibration* With this calibration procedure an estimation of the transformation from a coordinate system fixed to the camera, to the coordinate system in the robot end-effector is estimated [(see 1, 4, 5)]. In the hand-eye configuration, this transformation will be constant.

Of course, in any case, it is also needed to have the robot arm well-calibrated. This *robot calibration* procedure will allow the gripper position, relative to the robot base, to be accurately known.

2.1. Feature Extraction in the Planar Case

As said before, in the first approximation of our visual servoing implementation, the 2D case is considered. It means that object motion will be restricted to lay on a plane, and, accordingly, robot motion constrained to a plane, parallel to the first one, will be sufficient.

In this case, the object to be tracked can be as simple as a black circular spot on a white background. The planar image coordinates of the spot centroid are the only features needed. The transformation of these features to relative object-camera position is straight.

2.2. Feature Extraction in the General Case

The first extension of the previous case is to estimate the depth of the object measured from the camera lens. And, in the most general case, estimation of the relative object-camera orientation, is needed.

To recover the three dimensional information from the planar views given by the camera, a stereo-camera setup can be used, together with some matching processing. On the other hand, if only one camera is being used (as is the case of our configuration), it is possible to estimate the full object-camera pose, from the matching between the features extracted from the image and a geometric or CAD model of the object.

In our experiments the object to be used in this general case is a planar template with 25 circular, black spots, arranged in matrix form. The object's features to be extracted from the images are the centroid location of each spot. The 3D position of this centroids must be accurately known, relative to a local coordinate system fixed to the template. Hence, a *template calibration* procedure is also needed now. These calibrated locations will be referred later as $P_{w_i} = (x_{w_i}, y_{w_i}, z_{w_i})$ (with $i = 1..25$).

In order to estimate the relative camera-object pose, a linear-equation system has to be solved, the coefficients of which are the coordinates of P_{w_i} and their projections onto the image-plane: $P_i = (X_i, Y_i)$, and the unknowns of which are the desired transformation matrix components. For a detailed description of this solution, see [3, 4].

In industrial applications, however, the objects of interest will have a less convenient appearance, and corners, edges, holes, *etc.*, can be considered as features to be extracted. In the present work, this generalization has not been covered, but the stages of system modelling and control, which are the main subject of this paper, would be the same after the object-camera transformation has been estimated.

3. SYSTEM MODELLING

In this section, two levels in the control scheme will be distinguished:

- The lower level, which is determined by the robot manipulator used in our implementation. A convenient model for this level is first obtained.
- In a second stage, the complete system, including a model for the camera, the image capturing process and the robot model itself, is approximated.

3.1. The Robot Control Level

The robot manipulator used in our implementation is the PUMA 560. Corke [1] reports a model for visual control of this robot using direct access to the servomotors and encoders. In our case, the robot is commanded at a higher level, using the *ALTER* facility ([6]). This facility is intended for real-time path control of the robot, and the references are given in cartesian coordinates, so there is no need of inverse kinematics programming.

This low-level stage operates in open loop from the point of view of the programmer (however, the arm controller hardware itself implements a PID controller for each joint, after solving the inverse kinematics problem).

This control level is implemented as a small program in a personal computer, which reads the reference coming from the higher level, checks it for saturation, and handles the communication with the PUMA robot, sending the desired destination position, and receiving acknowledgement messages. From the point of view of systems modelling, this level has a sampling period $T_1 = 28$ ms, and deliberately introduces an integrator transfer function. This is justified because the higher level is implemented as a velocity control loop.

The destination position sent by this program is received by the PUMA controller module (VAL-II). Then, the following steps are carried out in this module:

- 37.5 ms after the starting of each cycle, the desired position is given to the servos. This step will be modeled as a pure transport delay.
- 27 ms after that, it is guaranteed the robot has reached this desired position. This will be modeled as a near straight-line response, with rising time 27 ms, approximated by the transfer function:

$$G_{\text{arm}}(s) = \frac{1}{(1 + \tau_1 s)(1 + \tau_2 s)} \quad \text{where: } \tau_1 = 0.003 \text{ and } \tau_2 = 0.005 \quad (1)$$

Then, the model used to approximate the behavior of each cartesian coordinate at the low control level is shown in Figure 2.

Several proofs have been made moving more than one axis at the same time, and it can be concluded that the six coordinates can be considered as independent control variables.

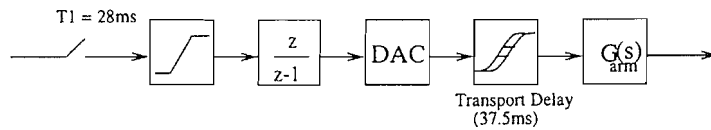


FIGURE 2 Detailed robot model.

3.2. The Visual Control Level

This is the high-level control stage. The sample time at this level, T_2 , is fixed between 3 and 9 times the low-level sampling period, depending on the complexity in the feature extraction procedure. It is important to note the multirate nature of this system.

Extending the one-dimensional model in Figure 2, to include both control levels, the block diagram showed in Figure 3 can be obtained. In this first model, as only one DOF is being used, the camera calibration procedure has been replaced by the estimation of a simple scale factor: K_l , *lens gain*, which converts distance in meters to distance in pixels.

Some indications about this diagram are given:

x_o represents the x -coordinate in the position of the object to be tracked, relative to the world reference system. This is compared with the position of the camera, x_c , getting the relative object-camera position, ${}^c x_o$.

The image acquisition and processing is modelled by a real-world to image-plane conversion module, and a time delay of one high-level period. The feature extracted from the image is the estimated, relative camera-object position, ${}^c \hat{x}_o$. Module (1) in the figure, is part of the image capture mechanism, and corresponds in this simplified case, to the real lens gain. Module (2), makes use of an estimation of the real lens gain value.

${}^c \hat{x}_o$ is compared to ${}^c \hat{x}_o^d$, the desired relative position. The necessary increment in the camera pose, Δx_c , is fed to the controller and, after the convenient transformation, the end-effector displacement is calculated. Finally, this is the value sent as a reference to the lower-level system.

Comparison experiments have proved the adequacy of this model to the real system. In order to make two comparison tests, the 1-DOF response of the system and the output of the model, under proportional control, were obtained. Figure 4a shows the comparison for

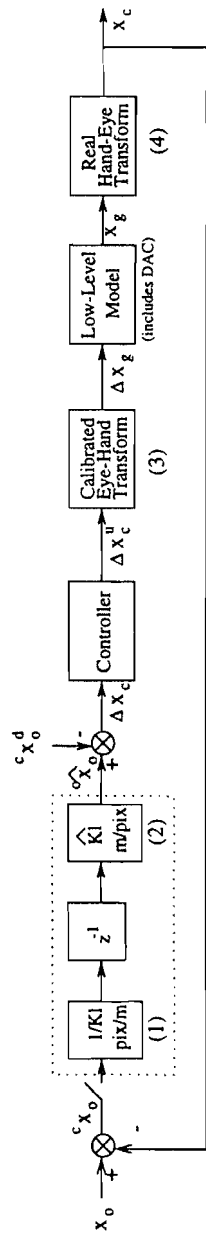


FIGURE 3 Detailed 1-DOF multirate model.

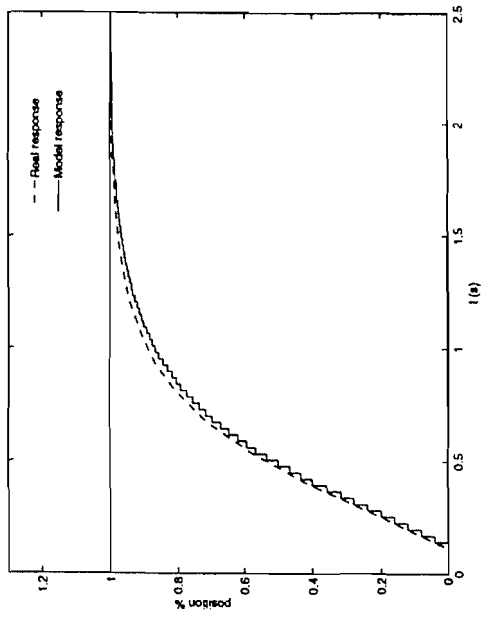
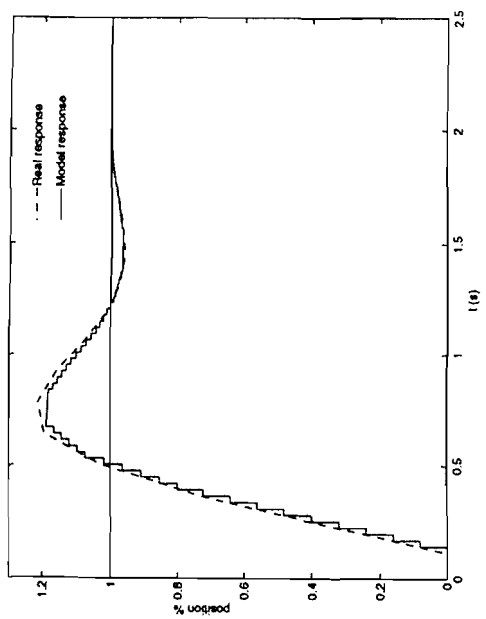


FIGURE 4 Response comparison under proportional control. (a) $K = 0.04$, (b) $K = 0.08$.

proportional gain $K = 0.04$, and Figure 4b for $K = 0.08$. In these experiments, the sampling times were: $T_1 = 28$ ms, $T_2 = 5 * T_1$.

This model will be used to evaluate the performance of the controllers to be designed before applying them to the real system.

3.3. Complete 6-DOF Scheme

In Figure 5, the extension of the previous model to 6 degrees of freedom can be seen. Where the common notation ${}^A H_B$, means for the homogeneous matrix that transforms coordinate system A into coordinate system B .

Again, a brief description of the modules in Figure 5 follows. The first symbolic difference operation, gives the relative pose of the camera and the object (where W is a stationary coordinate system, usually in the base of the robot), that, in matrix form, can be expressed as:

$${}^{C_{k+1}}H_{O_{k+1}} = [{}^W H_{C_{k+1}}]^{-1} \cdot {}^W H_{O_{k+1}}$$

The *World-to-Screen* module simulates the transformation made by the image capturing mechanism: it converts the P_{w_i} points to the corresponding P_i points (see Section 2).

The *Extrinsic Calibration* module gives the estimate object-camera pose, ${}^{C_k} \hat{H}_{O_k}$. Then the difference from the desired pose is calculated:

$$\Delta H_{C_k} = {}^{C_k} H_{C_{k+1}} = {}^{C_k} \hat{H}_{O_k} \cdot {}^{O_k} H_{C_{k+1}} = {}^{C_k} \hat{H}_{O_k} \cdot H_d^{-1}$$

Then, the camera-gripper transformation module is applied (note that ${}^C H_G$ is constant, as the camera is rigidly mounted on the robot gripper).

$$\Delta H_{G_k} = {}^{G_k} H_{G_{k+1}} = {}^G \hat{H}_C \cdot \Delta H_{C_k} \cdot {}^C \hat{H}_G$$

The controller module will have the general form: $x_k^u = a_1 x_{k-1}^u + \dots + a_n x_{k-n}^u + b_0 x_k + b_1 x_{k-1} + \dots + b_m x_{k-m}$. Where x can be replaced by each one of the six coordinates, extracted from the homogeneous matrix. The only consideration is that these coordinates must be referred to a stationary reference system, in order to the additivity be applicable from one cycle to another. The following transformation expresses the matrix ΔH_{G_k} respect to the world coordinate system:

$$\Delta H_{G_k}^{(W)} = {}^W \hat{H}_{G_k} \cdot \Delta H_{G_k} \cdot {}^{G_k} \hat{H}_W$$

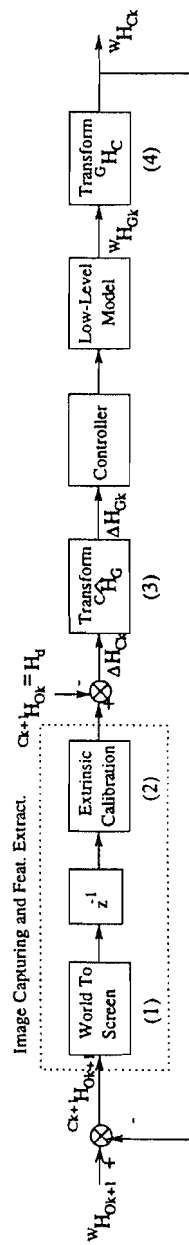


FIGURE 5 Complete 6-DOF model.

3.4. Disturbance Sources

From the block diagram shown in Figure 5, the main sources of disturbances can be pointed out:

- Modules (1) and (2) in Figure 5 are ideally complementary. In real circumstances, there is an error due mainly to:
 - The accuracy in the template calibration process and
 - The precision with which the centers of the circles are estimated. This depends on the frame-grabber resolution, camera-object distance, but overall to the velocity of the camera while the image is being grabbed. The effect of the velocity, introduces a deviation in the measure of the template centers, and will be studied in future work.
 - The calibrated internal camera parameters, as they appear in the extrinsic calibration equations.
 - The accuracy of the geometric model of the template.
- The same can be said regarding to modules (3) and (4). In this case, is the hand-eye calibration procedure what gives an estimation of the ${}^C H_G$ transformation.

4. CONTROLLER DESIGN AND EXPERIMENTAL RESULTS

4.1. Reduced Model

A detailed model for the visual servoing system at hand has been obtained. However, in order to make easier the controller design process, it is highly convenient to obtain a simplified model of our system.

The low-level system can be approximated by the block diagram shown in Figure 6. Where the saturation has not been considered. This

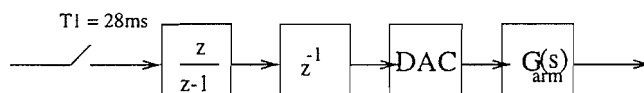


FIGURE 6 Reduced ALTER model.

is possible because it is imposed by software as a security speed limit, but high enough for not to take part in normal operation.

The ZOH equivalent of $G_{\text{arm}}(s)$ is obtained. Then, the resulting transfer function is:

$$G_{\text{robot}}(z) = \frac{1}{z-1} \cdot G_{\text{arm}}(z)$$

Finally, to unify our multi-rate system to just one sampling period, this transfer function, $G_{\text{robot}}(z)$ will be resampled at the high-level frequency, giving:

$$G'_{\text{robot}}(z) = \frac{4z+1}{z(z-1)}$$

Thus, the final reduced model is shown in Figure 7.

4.2. Performance of Different Controllers

Based on the one-dimensional reduced model just shown, a number of controllers have been designed and tested by simulation and real experiments. Only those controllers with best performance will be shown here:

The Step response of the different controllers will be compared with the best achievable response under proportional control:

- *Proportional Controller* First of all, a proportional controller was evaluated. It has been found that the gain value $K = 0.05$, gives the best performance under proportional control.

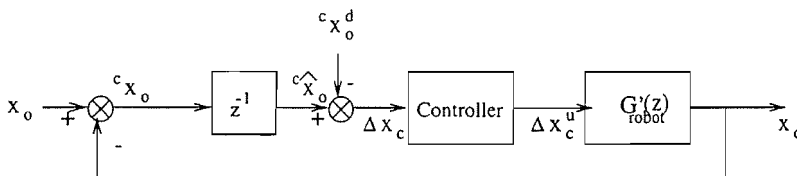


FIGURE 7 Approximated model used for controller design.

- *Pole-Placement Controller* A discrete controller with the general structure:

$$\frac{q_0 + q_1z^{-1} + q_2z^{-2}}{1 + p_1z^{-1} + p_2z^{-2}} \quad (2)$$

has been designed, using the pole-placement method. The specified poles for the closed-loop system were: $(z - 0.1)^5$. The response for this controller, compared with the proportional one just mentioned, can be seen in Figure 8.

- *Proportional Controller with Smith Predictor* In order to try to reduce the effects of the system delays, the Smith predictor is introduced. Figure 8 shows also the response for this controller. In this case the reference is reached after approximately the same time as for the previous controller, but a slight overshoot can be appreciated in case of the second controller.

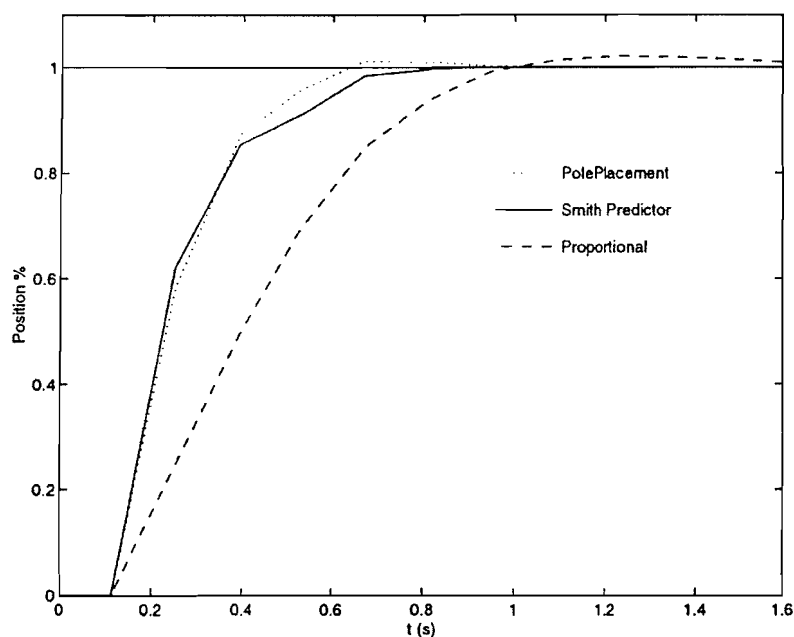


FIGURE 8 Closed-loop response for different controllers.

The PID controller with Smith predictor was also tested, but the results did not show any significant improvement.

5. IMPLEMENTATION

The visual servoing system implementation described in this chapter, is made of the following main components:

- *UNIMATION robot system* Composed by the VAL II controller module and the PUMA 560 robotic arm.
- *Pulnix CCD camera*.
- *Personal Computer* Pentium 160 MHz. Plugged into the PC, there is a multiple Digital Signal Processors Motherboard, with three DSPs modules, one of which is an image acquisition module. The parallelizing capabilities of this system have not been exploited yet.

The physical link between the personal computer and the robot controller is the serial line using the protocol RS-232 C.

6. CONCLUSIONS

This paper introduces a model for a visual servoing system, using the position based approach. The system is based on the PUMA 560 robot manipulator, operated using the ALTER facility. When operating in ALTER mode, the robot is commanded in cartesian positions; thus, it is not necessary to solve the inverse kinematics problem, and no direct control on the servomotors is available.

The obtained model (which happens to be a multirate one) has been validated in some experimental tests, comparing its response to the real response of the system.

Then, several controllers have been designed, starting from an approximated single-rate simplified model. The step response of the controllers with best performance is showed.

Current and future work is focused on the implementation of the extended 6-DOF visual servoing system, and on the generalization of the kind of features to be extracted for the object pose estimation. Besides this, a similar visual servoing system is being implemented on a mobile robot.

Acknowledgement

The authors would like to thank to CICYT for supporting this work under grant TAP 98-0541.

References

- [1] Corke, P. I. (1996). Visual Control of Robots. Research Studies Press Ltd.
- [2] Sanderson, A. C. and Weiss, L. E. (1980). Image-based Visual Servo Control Using Relational Graph Error Signals. *Proc. IEEE*, pp. 1074–1077.
- [3] Tsai, R. Y. and Lenz, R. K., A New Technique for Fully Autonomous and Efficient 3D Robotics Hand/Eye Calibration. *IEEE Transactions on Robotics and Automation*, 5(3), June, 1989.
- [4] Vargas, M. and Rubio, F. R., *DSP-Based Structure for Visual Tracking*. In: *Proceedings AARTC'97*. Vilamoura, Portugal, April, 1997.
- [5] Tsai, R. Y., A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Trans. on Robotics and Automation*, RA-3(4), August, 1987.
- [6] Unimate Industrial Robot. User's Guide to VAL II. Unimation. December, 1986.
- [7] Hashimoto, K. (Edited by) (1993). Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback. World Scientific.
- [8] Allen, P. K., Timcenko, A., Yoshimi, B. and Michelman, P., Automated Tracking and Grasping of a Moving Object with a Robotic Hand-Eye System. *IEEE Trans. on Robotic and Automation*, Vol. 9-2, 152–165, April, 1993.
- [9] Papanikolopoulos, N. P. and Khosla, P. K., Adaptive Robotic Visual Tracking: Theory and Experiments. *IEEE Transactions on Automatic Control*, 38(3), March, 1993.