2-D Visual Servoing for MARCO

Teresa A. Vidal C.
Institut de Robòtica i Informàtica Industrial
Universitat Politèctica de Catalunya - CSIC
Llorens i Artigas 4-6, Edifici U, 2a pl. Barcelona 08020, Spain
tvidal@iri.upc.es
IRI-DT-03-02. September 2003.

Abstract

This report presents an image-based visual servoing for our mobile platform called MARCO. The kinematic model of MARCO has been obtained in this work. An implementation has been developed using camera calibration parameters. Simulations and experimental results are presented here.

1 Introduction

The use of visual information in a feedback loop is an attractive solution for the position and motion control of autonomous mobile robots evolving in dynamic environments.
Visual servo systems typically use one or two camera configurations: end-effector mounted (eye-in hand), or fixed in the workspace.
Existing eye-in hand approaches are typically classified in two different categories: position based and image based control systems. In a position-based control system, the input is computed in the three-dimensional (3-D) Cartesian space. The pose of the target with respect to the coordinate system of the camera is estimated from image features corresponding to the perspective projection of the target in the image. In this approach knowledge of a perfect geometric model of the object is necessary.
Contrary, for an image-based control system, the input is computed in a 2-D image space (2-D Visual Servoing) [2]. The problem with 2-D visual servoing is in terms of image regulation, without an explicit 3-D reconstruction of the scene. A clear disadvantage of this kind of control system is that convergence is theoretically ensured only in a small region around the desired position.
To overcome such difficulty a more recent approach has been developed [5], the 2-1/2-D visual servoing. This control system is based on the estimation of the camera displacement (the rotation and the scaled translation of the camera) between the current and desired views of an object.
The work presented in this report is based on 2-D visual servoing for a wheeled mobile robot with an on-board camera mounted on a pan and tilt neck. The kinematic screw

between the camera and the scene is obtained by an image control law. However, it is still necessary to obtain the kinematic model of the robot and head in order to compute the linear and angular velocities of the mobile platform and the angular velocities of the pan and tilt.
This technical report is organized as follows, section two presents the kinematics of the complete mobile platform MARCO. In section three the perspective projection and way to model the image motion is derived. The feedback control law is presented in section four. Some simulations are shown in section five. In section six the technical features of the robot are described, along with a brief description of the application developed for the experiments results and conclusions are also presented.

2 Kinematic model of MARCO

Our mobile platform MARCO can be schematically represented, from a mechanical point of view, as a kinematic chain of rigid bodies (links) connected by joints. One end of the chain is constrained to the base, while an end effector (in this case the camera) is mounted to the other end. The resulting motion of the structures is obtained by composition of the elementary motions of each link with respect to the previous one. Using Denavit-Hartenberg convention it is easy to deduce the direct kinematics function of the complete chain. The construction of the direct kinematics is derived from the position and orientation change of the frames attached to the link (see Figure.1). The change of coordinates of the points in the camera coordinate frame with respect to the world coordinate frame is

$$M_W^c = M_W^r M_r^{pan} M_{pan}^{tilt} M_{tilt}^c.$$

We next describe each of these homogenous transformation matrix.

2.1 Robot Coordinate Frame ($\mathbf{S_r}$) - World Coordinate Frame ($\mathbf{S_w}$)

The homogenous transformation matrix (HTM) for the Robot Coordinate Frame to the World Frame is computed by,

$$M_W^r = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & X_r \\ \sin\theta_1 & \cos\theta_1 & 0 & Y_r \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where X_r and Y_r are the position of the center of the robot axle with respect to the World Coordinate Frame, and θ_1 is the angle between X_r and X_W about axis Z_W.

2.2 Pan Coordinate Frame ($\mathbf{S_{pan}}$) - Robot Coordinate Frame ($\mathbf{S_r}$)

The HTM from the Pan to the Robot is,

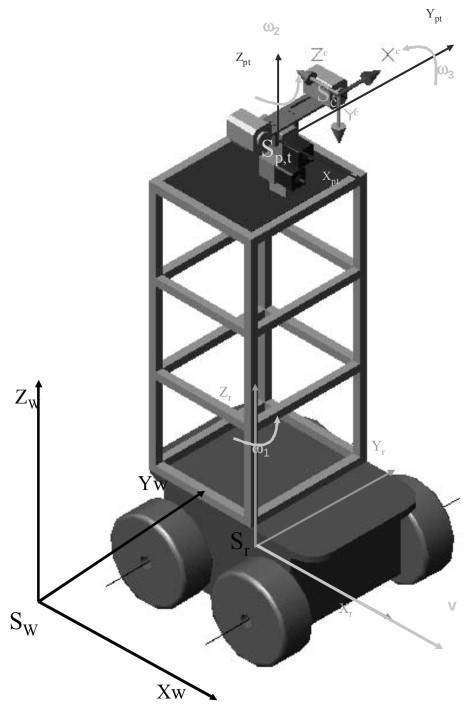

Figure 1: Mobile Robot MARCO and its Frame Coordinates.

$$M_r^{pan} = \begin{pmatrix} \cos\theta_2 & 0 & -\sin\theta_2 & DX \\ \sin\theta_2 & 0 & \cos\theta_2 & 0 \\ 0 & -1 & 0 & h \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where DX and h are shown in 2. The pan angle is θ_2.

2.3 Tilt Coordinate Frame ($\mathbf{S_{pan}}$)- Pan Coordinate Frame ($\mathbf{S_{tilt}}$)

$$M_{pan}^{tilt} = \begin{pmatrix} \cos\theta_3 & 0 & \sin\theta_3 & 0 \\ \sin\theta_3 & 0 & -\cos\theta_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where θ_3 is the tilt angle.

2.4 Tilt Coordinate Frame ($\mathbf{S_{tilt}}$)-Camera Coordinate Frame ($\mathbf{S_C}$)

It is necessary to find the HTM from the camera to the tilt M_{tilt}^c to obtain the relationship between camera and the World Frame.

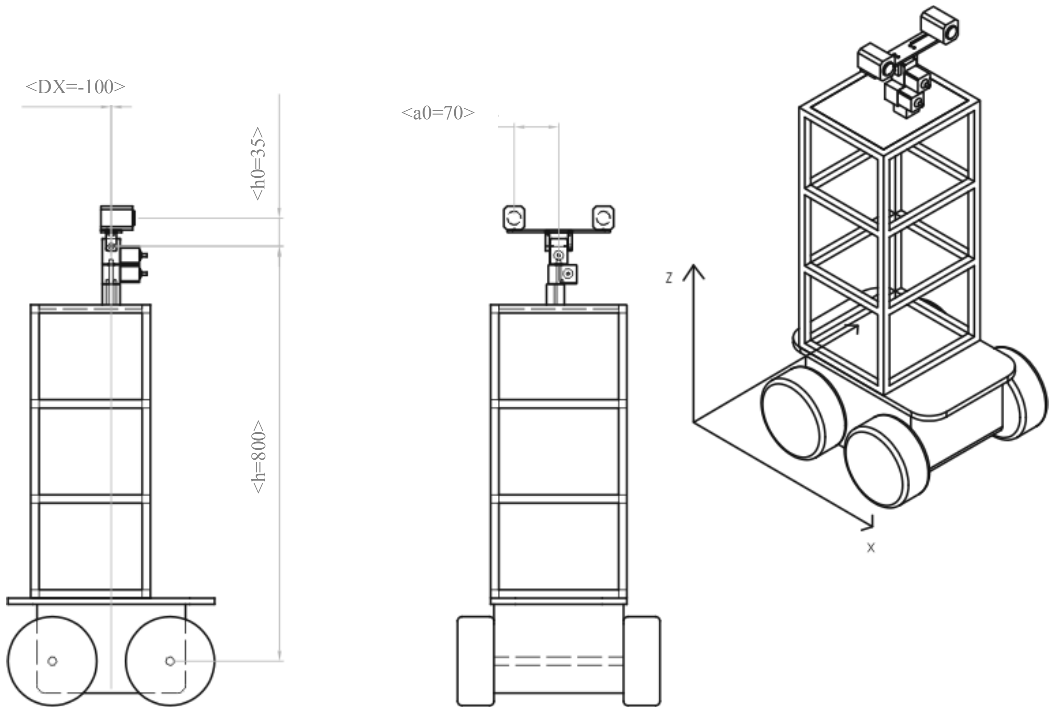

Figure 2:

$$M^c_{tilt} = \begin{pmatrix} 0 & 0 & 1 & b_0 \\ -1 & 0 & 0 & a_0 \\ 0 & -1 & 0 & h_0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b_0, a_0, and h_0 are the coordinates of the S_c center of projection in the camera frame (see 2). These parameters can be obtained by calibration of the neck-eye system.

2.5 Camera Coordinate Frame ($\mathbf{S_C}$) - Fixed Coordinate Frame ($\mathbf{S_w}$)

The full Transformation Matrix between the Camera Frame and the World Coordinate System is,

$$M^c_W = \begin{pmatrix} \sin(\theta_1+\theta_2) & -\sin\theta_3\cos(\theta_1+\theta_2) & \cos\theta_3\cos(\theta_1+\theta_2) & X_r + DX\cos\theta_1 - a_0\sin(\theta_1+\theta_2) + b_0\cos\theta_3\cos(\theta_1+\theta_2) + h_0\sin\theta_3\cos(\theta_1+\theta_2) \\ -\cos(\theta_1+\theta_2) & -\sin\theta_3\sin(\theta_1+\theta_2) & \cos\theta_3\sin(\theta_1+\theta_2) & Y_r + DX\sin\theta_1 + a_0\cos(\theta_1+\theta_2) + b_0\cos\theta_3\sin(\theta_1+\theta_2) + h_0\sin\theta_3\sin(\theta_1+\theta_2) \\ 0 & -\cos\theta_3 & -\sin\theta_3 & h + r - b_0\sin\theta_3 + h_0\cos\theta_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.6 Mobile Robot Kinematics

Most wheeled mobile robots have the non-holonomic property that the number of degrees of freedom of the input, linear $\mathbf{v}$ and angular ω_1 velocities, is less than that of the configuration (position (X_r, Y_r) and orientation θ_1). This relationship can be expressed as

$$\begin{pmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\theta}_1 \end{pmatrix} = \begin{pmatrix} \cos\theta_1 & 0 \\ \sin\theta_1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \end{pmatrix} \tag{1}$$

2.7 Camera Velocity

The origin of the Camera Frame is located at point Oc and the origin of the Robot Frame is located at Or Now its velocity can be decomposed in rotational and translational. The following camera velocity is expressed in terms of the robot frame.

$$\begin{aligned} V_{Oc} &= V_{Or} + \Omega_{S_r/S_W} \times \overrightarrow{OrOc} + V_{Oc/S_r} \\ \boldsymbol{\Omega}_{S_c/S_W} &= \Omega_{S_r/S_W} + \Omega_{S_c/S_r} \end{aligned}$$

2.8 Relationship between Kinematic Matrix and Robot

The pan and tilt mount vector velocity in terms of the World Frame S_W is

$$V_{S_{tilt}/S_W} = \left(\frac{\partial T_W^{tilt}}{\partial t} \right)$$

with

$$\begin{aligned} T_W^{tilt} &= \begin{pmatrix} X_r + DX \cos\theta_1 \\ Y_r + DX \sin\theta_1 \\ r + h \end{pmatrix} \\ \frac{\partial T_W^{tilt}}{\partial t} &= \begin{pmatrix} \dot{X}_r - \dot{\theta}_1 DX \sin\theta_1 \\ \dot{Y}_r + \dot{\theta}_1 DX \cos\theta_1 \\ 0 \end{pmatrix} \end{aligned} \tag{2}$$

Substituting the equation 1 in 9, we obtain

$$V_{S_{tilt}/S_W} = \underbrace{\begin{pmatrix} \cos\theta_1 & -DX\sin\theta_1 & 0\ 0 \\ \sin\theta_1 & DX\cos\theta_1 & 0\ 0 \\ 0 & 0 & 0\ 0 \end{pmatrix}}_{J_v} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \boldsymbol{\omega}_2 \\ \boldsymbol{\omega}_3 \end{pmatrix}$$

Let $\mathbf{\Omega}_{S_{tilt}/S_W}$ be the rotation vector of the pan and tilt mount with respect to the reference frame S_W, that is obtained as

$$\mathbf{\Omega}_{S_{tilt}/S_W} = \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \dot{\theta}_3 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \dot{\theta}_3 \\ \dot{\theta}_1 + \dot{\theta}_2 \end{pmatrix}$$

Rewriting the last equation in terms of the robot inputs,

$$\mathbf{\Omega}_{S_{tilt}/S_W} = \underbrace{\begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \\ 0\ 1\ 1\ 0 \end{pmatrix}}_{J_\Omega} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

The coordinate frames S_c and S_{tilt} are linked rigidly, so we can apply the kinematics equation of the rigid body:

$$V_{c/S_W} = V_{S_{tilt}/S_W} + \overrightarrow{OcOt} \times \Omega_{S_{tilt}/S_W}$$

where $\overrightarrow{OcOt}$ is the distance between the origin of the Tilt Frame and the origin of the Camera Frame referred to the World Coordinate System. The angular velocity is,

$$\Omega_{c/S_W} = \Omega_{S_c/S_W} = \Omega_{S_{tilt}/S_W}$$

The translation between the camera and the tilt is T^c_{tilt}, changing the reference to the World Frame results $\overrightarrow{OcOt} = -R^{tilt}_W T^c_{tilt}$ then

$$V_{c/S_W} = J_v \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} - R^{tilt}_W T^c_{tilt} \times J_\Omega \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}. \tag{3}$$

Defining the matrix,

$$\Delta = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}$$

where

$$\begin{aligned} a_1 &= -a_0 \sin(\theta_1 + \theta_2) + b_0 \cos\theta_3 \cos(\theta_1 + \theta_2) + h_0 \sin\theta_3 \cos(\theta_1 + \theta_2) \\ a_2 &= a_0 \cos(\theta_1 + \theta_2) + b_0 \cos\theta_3 \sin(\theta_1 + \theta_2) + h_0 \sin\theta_3 \sin(\theta_1 + \theta_2) \\ a_3 &= -b_0 \sin\theta_3 + h_0 \cos\theta_3 \end{aligned}$$

and $T^c_{tilt} = -\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}^T$, we can rewrite 3 as

$$R^{tilt}_m T^c_{tilt} \times J_\Omega \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \Delta J_\Omega \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \tag{4}$$

Because of 4 the Jacobian can be obtained as follows

$$\mathbf{J} = \begin{pmatrix} (J_v - \Delta J_\Omega) \\ J_\Omega \end{pmatrix},$$

or

$$\mathbf{J} = \begin{pmatrix} \cos\theta_1 & -a_2 - DX\sin\theta_1 & -a_2 & a_3 \\ \sin\theta_1 & a_1 + DX\cos\theta_1 & a_1 & 0 \\ 0 & 0 & 0 & -a_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

The relationship between the robot and camera velocities (expressed in the World Coordinate Frame) is shown by the following expression.

$$\begin{pmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \mathbf{J} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

Changing the last expression in terms of the camera frame, we have

$$\begin{aligned} V_{c/S_W}^{(S_c)} &= \left(R_{tilt}^{c}\right)^{-1}\left(R_{W}^{tilt}\right)^{-1} V_{c/S_W}^{(S_W)} \\ \Omega_{S_c/S_W}^{(S_c)} = \Omega_{S_{tilt}/S_W}^{(S_c)} &= \left(R_{tilt}^{c}\right)^{-1}\left(R_{W}^{tilt}\right)^{-1} \Omega_{S_{tilt}/S_W}^{(S_W)} \end{aligned}$$

Extending the last two equations:

$$V_{c/S_W}^{(S_c)} = \left(R_{tilt}^{c}\right)^{-1}\left(R_{W}^{tilt}\right)^{-1}\left[J_v \begin{pmatrix} \mathbf{v} \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} - R_W^{tilt} T_{tilt}^{c} \times J_\Omega \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}\right]$$

$$\Omega_{S_c/S_W}^{(S_c)} = \left(R_{tilt}^{c}\right)^{-1}\left(R_{W}^{tilt}\right)^{-1} J_\Omega \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}$$

the camera velocity in the camera reference frame stays as follow,

$$\begin{aligned} \mathbf{T}_c^{(S_c)} &= \begin{pmatrix} V_{c/S_W}^{(S_c)} \\ \Omega_{S_c/S_W}^{(S_c)} \end{pmatrix} \\ &= \mathbf{J}^{(S_c)} \begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \end{aligned}$$

where

$$\mathbf{J}^{(S_c)} = \begin{pmatrix} \left(R_{tilt}^{c}\right)^{-1}\left(R_{m}^{tilt}\right)^{-1}\left(J_v - \Delta J_\Omega\right) \\ \left(R_{tilt}^{c}\right)^{-1}\left(R_{m}^{tilt}\right)^{-1} J_\Omega \end{pmatrix}$$

and

$$\begin{aligned} \left(R_W^c\right)^{-1} &= \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} (\cos\theta_3)\cos(\theta_1+\theta_2) & (\cos\theta_3)\sin(\theta_1+\theta_2) & -\sin\theta_3 \\ -\sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 \\ (\sin\theta_3)\cos(\theta_1+\theta_2) & (\sin\theta_3)\sin(\theta_1+\theta_2) & \cos\theta_3 \end{pmatrix} \\ & \begin{pmatrix} \sin\theta_3\cos(\theta_1+\theta_2) & \sin\theta_3\sin(\theta_1+\theta_2) & \cos\theta_3 \\ -\cos\theta_3\cos(\theta_1+\theta_2) & -\cos\theta_3\sin(\theta_1+\theta_2) & \sin\theta_3 \\ \sin(\theta_1+\theta_2) & -\cos(\theta_1+\theta_2) & 0 \end{pmatrix} \end{aligned}$$

Computing the Jacobian of our robot in terms of the camera reference frame, we have:

$$\mathbf{J}^{(S_c)} = \begin{pmatrix} J_1 & J_2 & J_3 & J_4 \end{pmatrix}$$

$$J_1 = \begin{pmatrix} \cos\theta_2 \sin\theta_3 \\ -\cos\theta_2 \cos\theta_3 \\ \sin\theta_2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$J_2 = \begin{pmatrix} -a_0 \sin\theta_3 + DX \sin\theta_2 \sin\theta_3 \\ a_0 \cos\theta_3 - DX \cos\theta_3 \sin\theta_2 \\ -DX \cos\theta_2 - b_0 \cos\theta_3 - h_0 \sin\theta_3 \\ \cos\theta_3 \\ \sin\theta_3 \\ 0 \end{pmatrix}$$

$$J_3 = \begin{pmatrix} -a_0 \sin\theta_3 \\ a_0 \cos\theta_3 \\ -b_0 \cos\theta_3 - h_0 \sin\theta_3 \\ \cos\theta_3 \\ \sin\theta_3 \\ 0 \end{pmatrix}$$

$$J_4 = \begin{pmatrix} -b_0 \cos(\theta_1 + \theta_2) + a_0 \cos\theta_3 \sin(\theta_1 + \theta_2) \\ -h_0 \cos(\theta_1 + \theta_2) + a_0 \sin\theta_3 \sin(\theta_1 + \theta_2) \\ -b_0 \sin\theta_3 \sin(\theta_1 + \theta_2) + h_0 \cos\theta_3 \sin(\theta_1 + \theta_2) \\ \sin\theta_3 \sin(\theta_1 + \theta_2) \\ -\cos\theta_3 \sin(\theta_1 + \theta_2) \\ -\cos(\theta_1 + \theta_2) \end{pmatrix}$$

The Jacobian of the robot is the relationship between camera velocities and robot velocities.

3 Camera Kinematics

The camera model used is the full perspective pin-hole model. The corresponding transformation from camera frame to the image plane is through perspective projection
Let us consider a camera with its related frame S_c moving with respect to a reference frame S_W as in Figure 3. The camera interacts with a part of the environment with an associated frame at the image plane.
The coordinates of a point on the image plane are

$$\frac{u - u_0}{\alpha_u} = \frac{x}{z}$$
$$\frac{v - v_0}{\alpha_v} = \frac{y}{z}$$

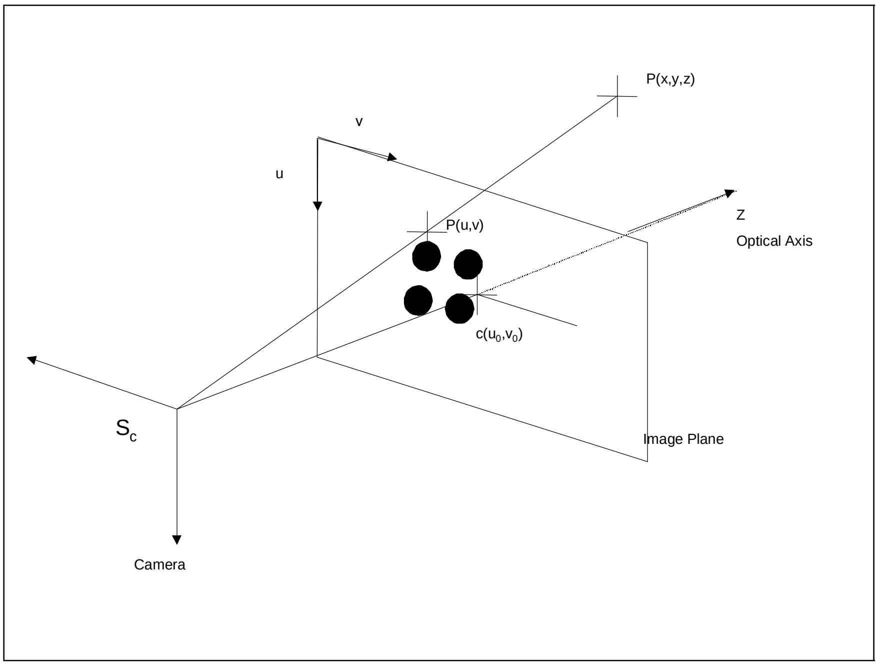

Figure 3:

where α_u, α_v, u_0 are v_0 intrinsic parameters of the camera. These parameters are obtained with camera calibration [6].
Image-based visual servoing is based on the selection in the image of a set s of visual features that has to reach a desired value s^*. Usually, s is composed of the image coordinates of several points belonging to the considered target. It is well known that the image Jacobian L_s^T plays a crucial role in the design of the possible control laws. This Jacobian relates the visual features velocity with the camera velocity [2].

$$\dot{s} = L_s^T T_c^{(Rc)} \tag{5}$$

where:
- $\dot{s}$ is the visual features velocity vector or the apparent motion between image pixels $\mathsf{dim}(m \times 1)$;
- L_s^T is the Image Jacobian or the interaction matrix $\mathsf{dim}(m \times 6)$ and has the information of the visual features s;
- $T_c^{(S_c)}$ is the the velocity vector of $\mathsf{dim}(6 \times 1)$ and represents the camera motion with respect to the scene. The elements $T_c = \begin{bmatrix} V_{c/S_r} & \Omega_{S_c/Sr} \end{bmatrix}^T$ are the translation and rotation velocities respectively, expressed in camera frame.
Using a classical perspective projection model with unit focal length, and if U and V coordinates of image points are selected in s, two successive rows of L_s^T are given by:

$$\begin{pmatrix} \dot{U} \\ \dot{V} \end{pmatrix} = \begin{pmatrix} \frac{1}{z} & 0 & \frac{U}{z} & UV & -(1+U^2) & V \\ 0 & \frac{1}{z} & \frac{V}{z} & (1+V^2) & -UV & -U \end{pmatrix} T_c^{(Rc)}$$

The depth z is used as $z = z_d$.

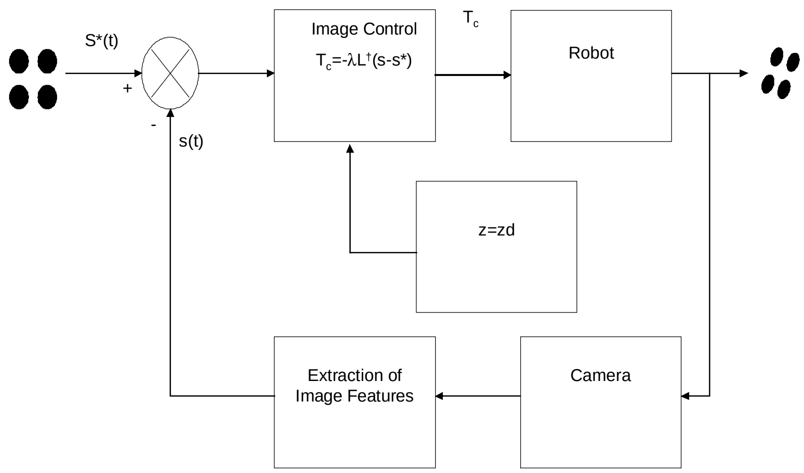

Figure 4: Block Diagram of the 2-D Visual Servoing

4 Image- Based Visual Servoing

Defining an error function as the difference between the current and desired images of the feature points

$$e = s - s^*,$$

the control law is defined as

$$\mathbf{T}_c^{(S_c)} = -\lambda \left(L_s^T\right)^\dagger e$$

where s^* is the desired position of the visual features. In our particular case is the position of each point of the pattern.[1]

The closed loop system stays in the following form,

$$\dot{s} = L_s^T \mathbf{T}_c^{(S_c)} = -\lambda L_s^T \left(L_s^T\right)^\dagger (s - s^*)$$

then if $L_s^T \left(L_s^T\right)^\dagger \geqslant \| s - s^* \|$ the system converges to the desired values.

Knowing $T_c^{(S_c)}$, it is possible to obtain the robot velocities, as described in the robot kinematics section, with

$$\begin{pmatrix} \mathbf{v} \\ \boldsymbol{\omega}_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \mathbf{J}^\dagger \mathbf{T}_c^{(S_c)}.$$

[1] Considering the † as the pseudo-ineverse. The same dimension as A^T so that $AXA = A$, $XAX = X$ and AX and XA are Hermitian.

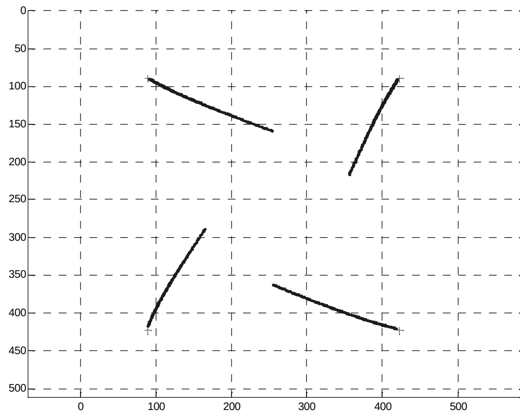

Figure 5: Image Space Trajectory.

In our case the interaction matrix has the following form,

$$L_s^T = \begin{pmatrix} \frac{1}{z^*} & 0 & \frac{U_1}{z^*} & U_1V_1 & -(1+U_1^2) & V_1 \\ 0 & \frac{1}{z^*} & \frac{V_1}{z^*} & (1+V_1^2) & -U_1V_1 & -U_1 \\ \frac{1}{z^*} & 0 & \frac{U_2}{z^*} & U_2V_2 & -(1+U_2^2) & V_2 \\ 0 & \frac{1}{z^*} & \frac{V_2}{z^*} & (1+V_2^2) & -U_2V_2 & -U_2 \\ \frac{1}{z^*} & 0 & \frac{U_3}{z^*} & U_3V_3 & -(1+U_3^2) & V_3 \\ 0 & \frac{1}{z^*} & \frac{V_3}{z^*} & (1+V_3^2) & -U_3V_3 & -U_3 \\ \frac{1}{z^*} & 0 & \frac{U_4}{z^*} & U_4V_4 & -(1+U_4^2) & V_4 \\ 0 & \frac{1}{z^*} & \frac{V_4}{z^*} & (1+V_4^2) & -U_4V_4 & -U_4 \end{pmatrix}$$

Figure 4 shows a block diagram of the 2-D visual servoing approach.

5 Simulations

The simulations presented here were developed in SIMULINK and the parameters are coincident with the real system. Figures 5, 6, 7 and 8 show some of the results obtained when applying image -based visual servoing to a robot like ours.
It is desirable not to saturate the actuators even in simulations, in Figure 7 it is shown how the velocities of the robot are reasonable for the real robot.
Figure 8 shows the error tending exponentially to zero.

6 Implementations and Experiments

A 2-D Visual Servoing implementation has been developed for the mobile robot MARCO. The application was created in Visual C++ using the Matrox Imaging Library for the segmentation, features extraction and image interpretation. Acquisition is made with a

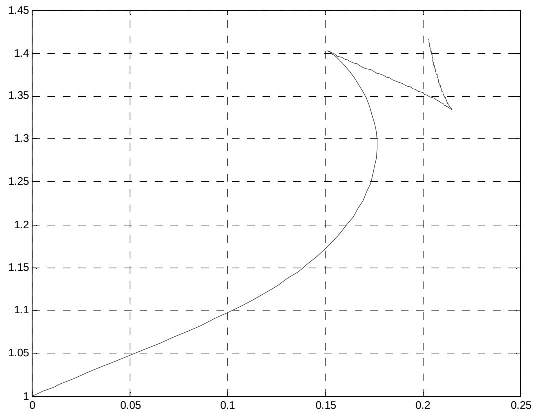

Figure 6: Workspace Trajectory.

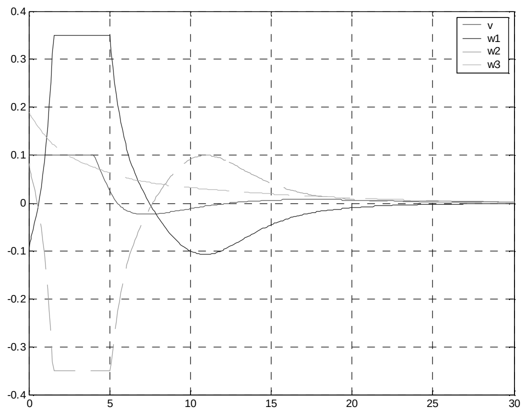

Figure 7: Robot Velocities.

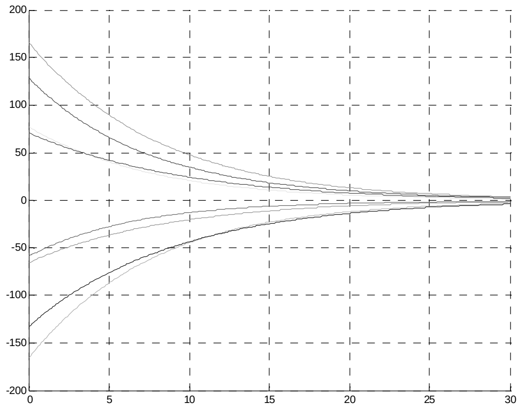

Figure 8: Errors (s-s*).

PXC200 frame grabber. The camera used is a SONY EVI-371DG with standard format PAL RGB. The robot MARCO has a mobile platform PIONEER 2.0. Image processing blocks are shown in Figure 9.
Saphira is the native operating software of the PIONEER Mobile Robots. Saphira can be thought of as having two architectures, one built on top of the other. The system architecture is an integrated set of routines for communicating and controlling the robot from a host computer. On top of the system routines is a robot control architecture, that is a design for controlling the mobile robot that addresses many of the problems involved in navigation, from low-level issues such as planning and object recognition [9].
The PXC200 acquires an image and it is stored in memory address, after that it is converted to a MIL image in order to be processed. The sequence followed is shown in Figure 9.
- Binarization (threshold 100);
- Filtering (noise removal);
- Region Segmentation (blobs) by the area and compacity;
- Getting blobs;
- Extraction of the center of each blob;
- Ordering of blobs.

In Figure 10 a segmented image is shown, with the corresponding blobs found and ordered.
The Visual C++ application shows at the screen the linear and angular instant velocities of the robot, the image space errors, the coordinates in millimeters of each center blobs, the frames per second of a refresh image and the segmented image as Figure 11 shows.
The sample time is about 60 ms, in consequence it is difficult to stabilize the system. Sometimes the pattern gets out of the image and it becomes impossible to recover control, then a null command of velocities is sent to the robot.
As it is mentioned before for this implementation, the depth is not calculated, instead we used the desired depth.
A disadvantage of this image-based visual servoing is the impossible trajectories for the

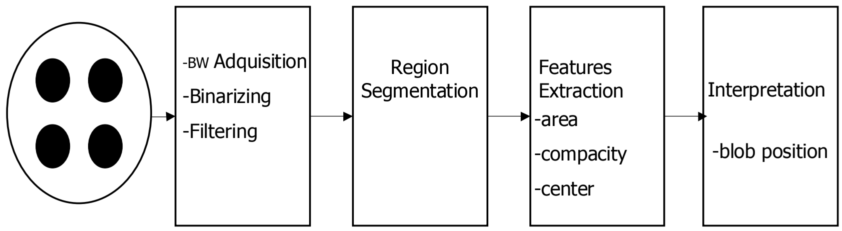

Figure 9: Image Procesing Block Diagram.

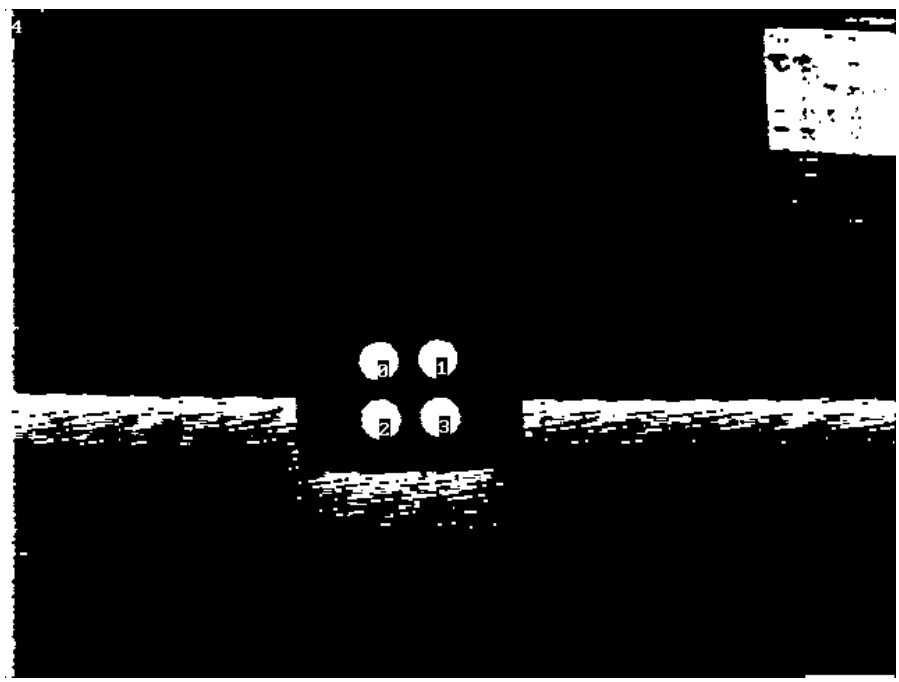

Figure 10: Segmented Image.

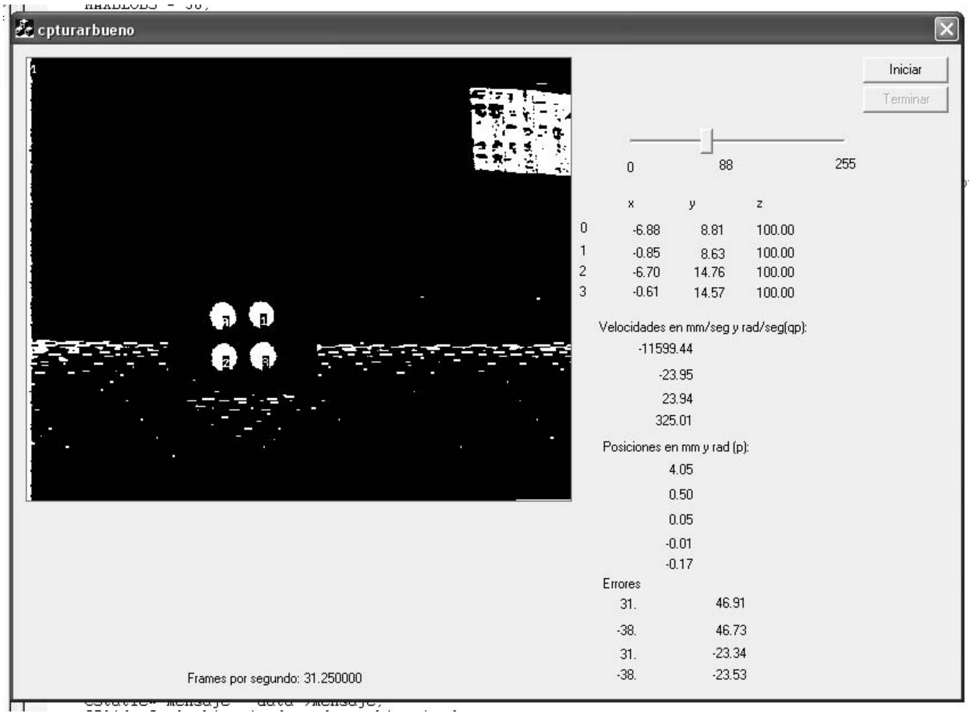

Figure 11: 2D Visual Servoing Aplication.

robot that the control law can compute. That is the reason why the 2 1/2 Visual Servoing was developed, but it becomes necessary to compute the depth.

References

[1] Cadenat V., R. Swain., P. Spuères, M. Devy (1999), A Controller to Perform a Visually Guided Tracking Task in Cluttered Enviroment.*Proc. of the IEEE/RSJ Intern. Conference on Intelligent Robots and Systems*, pp. 775-780.

[2] Chaumette F., Rives P., Espiau B. The Task Function Approach Applied to Vision-Based Control. *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pp 1392-1397.

[3] Swain-Oropeza R. Contrôle de tâches référencées vision pour la navigation d'un robot mobile en milieu structuré. *L'Institut National Polytechnique de Toulouse PhD Thesis,* , 1999.

[4] Hutchinson Seth, Hager G. D., Corke P.I. A tutorial on Visual Servo Control. *IEEE Transactions on Robotics and Automation, Vol. 12, No 5, October 1996,* pp.651-670.

[5] Malis, E.. Contributions à la modélisation et à la commande en asservissement visuel. *Université de Rennes I PhD Thesis.* 1998.

[6] Andrade, J. Camera Calibration. *Technical Report IRI- DT 2001/2,* Juny 2001.

[7] Sciavicco L., Siciliano B. Modelling and Control Robot Manipulators. Chapters on Kinematics and Differential Kinematics and Statitcs.*Springer-Verlag* 2000.

[8] Zhang H., Ostrwski J.P. Visual Motion Planning for Mobile Robots. *IEEE Transactions on Robotics and Automation,* Vol. 18, No2, April 2002.

[9] Konolige K. SAPHIRA Sofware Manual. Version 6.1.1998.