Stable Visual Servoing of Camera-in-Hand Robotic Systems

Rafael Kelly, *Member, IEEE*, Ricardo Carelli, *Senior Member, IEEE*, Oscar Nasisi, Benjamín Kuchen, *Member, IEEE*, and Fernando Reyes

***Abstract*—In this paper, the control problem of camera-in-hand robotic systems is considered. In this approach, a camera is mounted on the robot, usually at the hand, which provides an image of objects located in the robot environment. The aim of this approach is to move the robot arm in such a way that the image of the objects attains the desired locations. We propose a simple image-based direct visual servo controller which requires knowledge of the objects' depths, but it does not need to use the inverse kinematics and the inverse Jacobian matrix. By invoking the Lyapunov direct method, we show that the overall closed-loop system is stable and, under mild conditions on the Jacobian, local asymptotic stability is guaranteed. Experiments with a two degrees-of-freedom direct-drive manipulator are presented to illustrate the controller's performance.**

***Index Terms*—Control of robots, Lyapunov stability, regulation, transpose Jacobian control, visual servoing.**

I. Introduction

The USE of visual information in the feedback loop is an attractive solution for the position and motion control of autonomous robot manipulators evolving in unstructured environments [1]. This robot control strategy, so-called visual servoing, can be classified in two approaches: fixed camera and camera in hand. In fixed-camera robotic systems, multiple cameras fixed in the world–coordinate frame capture images of both the robot and its environment. The objective of this approach is to make the robot move in such a way that its end effector reaches a desired object [2]–[8]. In the camera-in-hand configuration, a camera is mounted on the robot, which supplies visual information of the environment, as depicted in Fig. 1. The objective of this approach is to move the manipulator in such a way that the projection of either a moving or a static object be always at a desired location in the image captured by the camera [9]–[18].

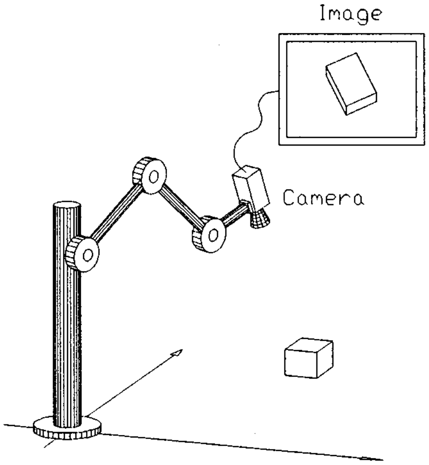


Fig. 1. Camera-in-hand robotic system.

This paper deals with the camera-in-hand approach to vision robot control. This control problem has attracted the attention of researchers in recent years (see [1] for an interesting historical review). A common characteristic of most of previous works is that the robot manipulator is assumed to be a positioning device whose dynamics do not interact with the visual feedback loop. Although this assumption is valid for slow robot motion, it does not hold for high-speed tasks where the robot dynamics are not neglectable, which is particularly true for direct-drive robots. One exception where the robot dynamics has been taken into account is the controller proposed by Hashimoto and Kimura [16]; a model-based direct visual robot controller considering static objects was designed following the input–output linearization technique, which leads to a control law dependent on the full nonlinear robot dynamic model.

In this paper, we are interested in position control (set-point regulation) in an environment with static objects. Focusing on this formulation, we attempt to design a simple controller supported by a rigorous stability analysis incorporating the full Lagrangian robot dynamics.

Our main theoretical contribution is the extension of the transpose Jacobian control philosophy [19] to image-based direct visual servo control for camera-in-hand robot manipulators. In this approach, the control aim is defined in terms of an image feature error and the robot joint torques are directly computed as the control actions. Thanks to this extension, a very simple image-based direct visual servo controller for camera-in-hand robotic systems is designed where the control algorithm does not need the solution of either the inverse image problem or the robot kinematics, and its structure is independent of the robot inertia and Coriolis matrices. Furthermore, closed-loop stability including the full nonlinear robot dynamics is rigorously proven by invoking the Lyapunov's direct method. In order to show local asymptotic stability, a weak assumption is imposed on the

Manuscript received February 25, 1997; revised May 15, 1998. Recommended by Technical Editor T. Fukuda. This work was supported in part by CONACyT (Mexico) and CONICET (Argentina), by the project *Perception Systems for Robots-CYTED,* and by CONACyT-NSF Grant 228050-5-C084A and Grant IRI-9613737.

R. Kelly is with the División de Física Aplicada, Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, B.C., 22800 Mexico (e-mail: rkelly@cicese.mx).

R. Carelli, O. Nasisi, and B. Kuchen are with the Instituto de Automática, Universidad Nacionál de San Juan, 5400 San Juan, Argentina.

F. Reyes is with the Universidad A. de Puebla, Puebla, Pue., 72001 Mexico.

Publisher Item Identifier S 1083-4435(00)02471-6.

1083–4435/00$10.00 © 2000 IEEE

Jacobian. This paper continues the work originally presented in [20] and followed by [21].

The proposed direct visual servo controller has been experimentally tested on a two-degree-of-freedom direct-drive vertical arm. We describe the experimental results obtained considering one and two object feature points.

Throughout this paper, we use the notations $\lambda_m\{A\}$ and $\lambda_M\{A\}$ to indicate the smallest and largest eigenvalues, respectively, of a symmetric positive-definite bounded matrix $A(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathbb{R}^n$. The norm of vector $\boldsymbol{x}$ is defined as $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$.

The paper is organized as follows. In Section II, we present the robotic system model and the control problem formulation. The proposed visual controller is introduced and analyzed in Section III. In Section IV, we present experimental results and, finally, in Section V, we offer some concluding remarks.

II. Robotic System Model and Control Problem Formulation

The robotic system considered in this paper is composed of a robot manipulator featuring a camera in its hand, as depicted in Fig. 1. The basic mathematical description of this system consists of the robot dynamics and differential kinematics and the camera model.

A. Robot Dynamics

In the absence of friction or other disturbances, the dynamics of a serial n-link rigid robot manipulator can be written as [22]

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{1}$$

where

$\boldsymbol{q}$	$n \times 1$ vector of joint displacements;
$\boldsymbol{\tau}$	$n \times 1$ vector of applied joint torques;
$M(\boldsymbol{q})$	$n \times n$ symmetric positive definite manipulator inertia matrix;
$C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$	$n \times 1$ vector of centripetal and Coriolis torques;
$\boldsymbol{g}(\boldsymbol{q})$	$n \times 1$ vector of gravitational torques.

One important property of the robot dynamics is the following.

Property 1—see, e.g., [22] and [23]: The time derivative of the inertia matrix, and the centripetal and Coriolis matrix satisfy

$$\dot{\boldsymbol{q}}^T \left[\tfrac{1}{2}\dot{M}(\boldsymbol{q}) - C(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]\dot{\boldsymbol{q}} = 0, \qquad \forall \boldsymbol{q}, \dot{\boldsymbol{q}} \in \mathbb{R}^n. \tag{2}$$

$\diamondsuit$

B. Robot Differential Kinematics

The differential kinematics of a manipulator gives the relationship between the joint velocities $\dot{\boldsymbol{q}}$ and the corresponding end-effector translational $\boldsymbol{v}_c$ and angular velocity $\boldsymbol{\omega}_c$. They are related via the *geometric* Jacobian $J_g(\boldsymbol{q})$ [24]

$$\begin{bmatrix} \boldsymbol{v}_c \\ \boldsymbol{\omega}_c \end{bmatrix} = J_g(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{3}$$

If the end-effector pose (position and orientation) is expressed with reference to a minimal representation in the operational space, it is possible to compute the Jacobian matrix via differentiation of the direct kinematics with respect to the joint positions. The resulting Jacobian, termed *analytical* Jacobian $J_A(\boldsymbol{q})$, is related to the geometric Jacobian by [24]

$$J_g(\boldsymbol{q}) = \begin{bmatrix} I & 0 \\ 0 & T(\boldsymbol{q}) \end{bmatrix} J_A(\boldsymbol{q}) \tag{4}$$

where $T(\boldsymbol{q})$ is a transformation matrix that depends on the parameterization of the end-effector orientation.

C. Model of the Camera

Let us consider a TV camera mounted at the robot end effector. The origin of the camera coordinate frame (end-effector frame) with respect to the robot coordinate frame is represented by $\boldsymbol{s}_c = \boldsymbol{s}_c(\boldsymbol{q}) \in \mathbb{R}^{m_0}$ with $m_0 = 3$. The orientation of the camera frame with respect to the robot frame is denoted by $R_c^T = R_c(\boldsymbol{q})^T \in SO(3)$. The image acquired by the camera supplies a two-dimensional (2-D) array of brightness values from a three-dimensional (3-D) scene. This image may undergo various types of computer processing to enhance image properties and extract image features. In this paper, we assume that the image features are the projection into the 2-D image plane of 3-D points in the scene space.

We also assume a perspective projection with a focal length λ, as depicted in Fig. 2. An object (feature) point with coordinates $[X \quad Y \quad Z]^T \in \mathbb{R}^3$ in the camera frame projects onto a point on the image plane, with image coordinates $[x \;\; y]^T \in \mathbb{R}^2$. The position $\boldsymbol{\xi} = [x \quad y]^T \in \mathbb{R}^2$ of an object feature point in the image will be referred to as an *image feature point* [11]. In this paper, we assume that the object can be characterized by a set of feature points. For the sake of completeness, in the following subsections we recall some concepts concerning single and multiple feature points.

1) Single Feature Point: Following the notation of [16], let $\boldsymbol{s}_o \in \mathbb{R}^{m_0}$ be the position of an object feature point expressed in the robot coordinate frame. Therefore, the relative position of this object feature located in the robot workspace with respect to the camera coordinate frame is $[X \;\; Y \quad Z]^T$. According with the perspective projection [1], the image feature point depends uniquely on the object feature position $\boldsymbol{s}_o$ and the camera position and orientation, and it is given by

$$\boldsymbol{\xi} = \begin{bmatrix} x \\ y \end{bmatrix} = -\alpha \frac{\lambda}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{5}$$

where α is the scaling factor in pixels per meter due to the camera sampling and $Z < 0$. This model is also called the imaging model [16]. The time derivative yields

$$\dot{\boldsymbol{\xi}} = -\frac{\alpha\lambda}{Z} \begin{bmatrix} 1 & 0 & -\dfrac{X}{Z} \\ 0 & 1 & -\dfrac{Y}{Z} \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}. \tag{6}$$

On the other hand, the position of the object feature point $\boldsymbol{s}_o$ with respect to the camera frame is given by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = R_c(\boldsymbol{q})[\boldsymbol{s}_o - \boldsymbol{s}_c(\boldsymbol{q})]. \tag{7}$$

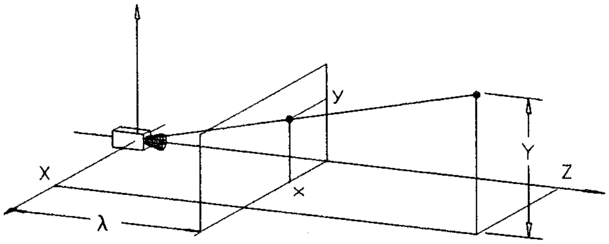


Fig. 2. A simple camera model.

By invoking the general formula for velocity of a moving point in a moving frame with respect to a fixed frame [29], the time derivative of (7) can be expressed in terms of the camera translational and angular velocities as [10]

$$\begin{bmatrix}\dot X\\ \dot Y\\ \dot Z\end{bmatrix} = \begin{bmatrix}-1&0&0&0&-Z&Y\\0&-1&0&Z&0&-X\\0&0&-1&-Y&X&0\end{bmatrix} \cdot \begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_c(\boldsymbol{q})\\ \boldsymbol{\omega}_c(\boldsymbol{q})\end{bmatrix} \tag{8}$$

where $\boldsymbol{v}_c$ and $\boldsymbol{\omega}_c$ stand for the camera translational and angular velocities with respect to the robot frame, respectively.

The motion of the image feature point as a function of the camera velocity is obtained by substituting (8) into (6)

$$\dot{\boldsymbol{\xi}} = -\frac{\alpha\lambda}{Z}\begin{bmatrix}-1&0&\frac{X}{Z}&\frac{XY}{Z}&-\frac{Z^2+X^2}{Z}&Y\\0&-1&\frac{Y}{Z}&\frac{Z^2+Y^2}{Z}&-\frac{XY}{Z}&-X\end{bmatrix} \cdot \begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_c(\boldsymbol{q})\\ \boldsymbol{\omega}_c(\boldsymbol{q})\end{bmatrix}. \tag{9}$$

Instead of using the coordinates X and Y of the object feature $\boldsymbol{s}_o$ described in the camera coordinate frame, which are *a priori* unknown, it is usual to replace them for the coordinates x and y of the projection of such a feature point onto the image frame. Therefore, by using (5) we get

$$\dot{\boldsymbol{\xi}} = J_{\text{image}}(\boldsymbol{\xi}, Z)\begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_c(\boldsymbol{q})\\ \boldsymbol{\omega}_c(\boldsymbol{q})\end{bmatrix} \tag{10}$$

where $J_{\text{image}}(\boldsymbol{\xi}, Z)$ is the so-called image Jacobian defined by [10]

$$J_{\text{image}}(\boldsymbol{\xi}, Z) = \begin{bmatrix}\frac{\alpha\lambda}{Z}&0&\frac{x}{Z}&-\frac{xy}{\alpha\lambda}&\frac{\alpha^2\lambda^2+x^2}{\alpha\lambda}&y\\0&\frac{\alpha\lambda}{Z}&\frac{y}{Z}&-\frac{\alpha^2\lambda^2+y^2}{\alpha\lambda}&\frac{xy}{\alpha\lambda}&-x\end{bmatrix}. \tag{11}$$

Finally, by using (3) and (4) we can express $\dot{\boldsymbol{\xi}}$ in terms of the robot joint velocity $\dot{\boldsymbol{q}}$ as

$$\begin{aligned}\dot{\boldsymbol{\xi}} &= J_{\text{image}}(\boldsymbol{\xi}, Z)\begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix}J_g(\boldsymbol{q})\dot{\boldsymbol{q}}\\ &= J_{\text{image}}(\boldsymbol{\xi}, Z)\begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix}\begin{bmatrix}I&0\\0&T(\boldsymbol{q})\end{bmatrix}J_A(\boldsymbol{q})\dot{\boldsymbol{q}}.\end{aligned} \tag{12}$$

2) Multiple Feature Points: In applications to objects located in a 3-D space, three or more feature points are required for the visual servo control to be solvable [14], [25]. The above imaging model can be extended to a static object located in the robot workspace having p object features points. In this case, $\boldsymbol{s}_o \in \mathbb{R}^{p\cdot m_0}$ is a constant vector which contains the p object feature points, and the feature image vector $\boldsymbol{\xi} \in \mathbb{R}^{2p}$ is redefined as

$$\boldsymbol{\xi} = \begin{bmatrix}x_1\\y_1\\ \vdots\\x_p\\y_p\end{bmatrix} = -\alpha\lambda\begin{bmatrix}\frac{X_1}{Z_1}\\ \frac{Y_1}{Z_1}\\ \vdots\\ \frac{X_p}{Z_p}\\ \frac{Y_p}{Z_p}\end{bmatrix} \in \mathbb{R}^{2p}.$$

The *extended* image Jacobian $J_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z}) \in \mathbb{R}^{2p\times m_0}$ is given by

$$J_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z}) = \begin{bmatrix}J_{\text{image}}\left(\begin{bmatrix}x_1\\y_1\end{bmatrix}, Z_1\right)\\ \vdots\\ J_{\text{image}}\left(\begin{bmatrix}x_p\\y_p\end{bmatrix}, Z_p\right)\end{bmatrix} \tag{13}$$

where $\boldsymbol{Z} = [Z_1 \;\; Z_2 \;\; \cdots \;\; Z_p]^T \in \mathbb{R}^p$.

Using (12) and (13), the time derivative of the image feature vector can be expressed as

$$\dot{\boldsymbol{\xi}} = J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})\dot{\boldsymbol{q}} \tag{14}$$

where

$$J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z}) = J_{\text{image}}(\boldsymbol{\xi}, \boldsymbol{Z})\begin{bmatrix}R_c(\boldsymbol{q})&0\\0&R_c(\boldsymbol{q})\end{bmatrix} \cdot \begin{bmatrix}I&0\\0&T(\boldsymbol{q})\end{bmatrix}J_A(\boldsymbol{q}) \in \mathbb{R}^{2p\times n} \tag{15}$$

will be called the Jacobian matrix hereafter in this paper.

D. Control Problem Formulation

The robot task is specified in the image plane in terms of the image feature values corresponding to the relative robot and object positions. Let us denote with $\boldsymbol{\xi}_d \in \mathbb{R}^{2p}$ the *desired image feature* vector which is assumed to be constant. For some tasks, the desired feature vector $\boldsymbol{\xi}_d$ can be obtained directly in the image feature space. Another way to get $\boldsymbol{\xi}_d$ is by using a "teach-by-showing" strategy [9]. In this approach, an image is transduced at the desired reference position and the corresponding extracted features represent the desired feature vector $\boldsymbol{\xi}_d$. The control problem is to design a controller which computes the applied torques $\boldsymbol{\tau}$ to move the robot in such a way that the actual image features reach the prescribed desired ones. The *image feature error* defined as $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}_d - \boldsymbol{\xi}$ may be calculated at every measurement time and used to drive the robot in a direction which decreases the error. Therefore, the control aim is to ensure that

$$\lim_{t\to\infty} \tilde{\boldsymbol{\xi}}(t) = \mathbf{0} \in \mathbb{R}^{2p}$$

provided that the initial feature error $\tilde{\boldsymbol{\xi}}(0)$ and joint velocity $\dot{\boldsymbol{q}}(0)$ are sufficiently small.

We make the following assumptions:

A0 The object is static, i.e., $\dot{\boldsymbol{s}}_o = \mathbf{0}$.

A1 There exists a robot joint configuration $\boldsymbol{q}_d$ for which the feature error vanishes, i.e., $\boldsymbol{\xi}_d = \boldsymbol{\xi}(\boldsymbol{q}_d)$. It should be pointed out that $\boldsymbol{q}_d$ does not need to be known nor measured.

A2 The unknown desired joint position $\boldsymbol{q}_d$ is an isolated solution of $\boldsymbol{\xi}(\boldsymbol{q}) = \boldsymbol{\xi}_d$.

A3 The Jacobian $J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))$ is continuously differentiable with respect to each entry of $\boldsymbol{q}$ and

$$\operatorname{rank}\{J(\boldsymbol{q}_d, \boldsymbol{\xi}(\boldsymbol{q}_d), \boldsymbol{Z}(\boldsymbol{q}_d))\} = n.$$

A4 The distance $\boldsymbol{Z}$ from the camera to the object (depth) is known.

Assumption A1 ensures that the control problem is solvable. Assumption A2 is a necessary condition to establish stability in an asymptotic sense. This is equivalent to assuming that the robot does not have kinematic redundancy with respect to the visual task. Assumption A3 is required for technical reasons in the stability analysis. A necessary condition to fulfill assumption A3 is to select the number of object points such that $2p \geq n$ and, hereafter, we suppose that this is the case. Concerning assumption A4, a practical way to obtain the depth $\boldsymbol{Z}$ is by using external sensors as ultrasound or additional cameras in the so-called binocular stereo approach [6], [7]. Other schemes use images obtained from different (more than two) points in the called active monocular stereo approach, and least-square techniques [10].

III. Simple Visual Servo Controller

In the control problem formulation considered in this paper, the position of the feature points of the object can only be measured through the camera, thus, a direct knowledge of the desired joint position $\boldsymbol{q}_d$ is not available. Notwithstanding, the desired joint position $\boldsymbol{q}_d$ can be obtained by solving the inverse image and kinematics problems. However, in this paper, we consider a simpler approach, which consists of directly using the image feature error $\tilde{\boldsymbol{\xi}}$ supplied in the image coordinate frame.

The approach followed in this paper was motivated by the transpose Jacobian control philosophy introduced in [19] (further analysis and discussions about this control scheme are in [26]–[34]).

The control law of the proposed controller is given by

$$\boldsymbol{\tau} = J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p \tilde{\boldsymbol{\xi}} - K_v \dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{16}$$

where $K_p \in \mathbb{R}^{2p\times 2p}$ and $K_v \in \mathbb{R}^{n\times n}$ are the symmetric positive-definite proportional and derivative matrices which are chosen by the designer. Fig. 3 depicts a closed-loop block diagram.

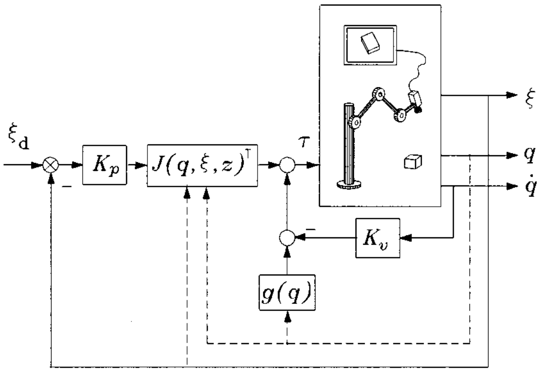


Fig. 3. Block diagram.

It is worth noticing that the controller uses directly the feature error vector $\tilde{\boldsymbol{\xi}}$ which is the difference between the desired feature vector and the actual one expressed in the image coordinate frame. The controller also requires the measurement of the joint position $\boldsymbol{q}$ and velocity $\dot{\boldsymbol{q}}$, the knowledge of the Jacobian matrix $J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})$, and the gravitational torque vector $\boldsymbol{g}(\boldsymbol{q})$. However, the solution of the inverse image and kinematics are obviated.

The closed-loop system is obtained by substituting the control action $\boldsymbol{\tau}$ from the control law (16) into the robot dynamics (1)

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p \tilde{\boldsymbol{\xi}} - K_v \dot{\boldsymbol{q}}. \tag{17}$$

The system behavior can be written in terms of the state vector $[\boldsymbol{q}^T \;\; \dot{\boldsymbol{q}}^T]^T \in \mathbb{R}^{2n}$ as

$$\frac{d}{dt}\begin{bmatrix}\boldsymbol{q}\\ \dot{\boldsymbol{q}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}}\\ M(\boldsymbol{q})^{-1}\Big[J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p \tilde{\boldsymbol{\xi}} - K_v \dot{\boldsymbol{q}} - C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}\Big]\end{bmatrix}. \tag{18}$$

Notice that the closed-loop system is described by an autonomous nonlinear differential equation and from assumption A1 we have that $[\boldsymbol{q}^T \;\; \dot{\boldsymbol{q}}^T]^T = [\boldsymbol{q}_d^T \;\; \mathbf{0}]^T \in \mathbb{R}^{2n}$ is an equilibrium point, regardless of whether or not the Jacobian $J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})$ is a square matrix.

Before presenting the stability analysis, we digress momentarily to establish the following.

Lemma 1: Under assumptions A0–A4, the equation

$$J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))^T K_p[\boldsymbol{\xi}_d - \boldsymbol{\xi}(\boldsymbol{q})] = \mathbf{0}$$

has an isolated solution at $\boldsymbol{q} = \boldsymbol{q}_d$. ∇

Proof: From assumptions A1 and A3, we have that $\operatorname{rank}\{J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))\} = n$ for all $\boldsymbol{q}$ in a neighborhood of $\boldsymbol{q}_d$. Using this fact and considering that K_p is a square nonsingular matrix ($\operatorname{rank}\{K_p\} = 2p$), then we have that $\operatorname{rank}\{J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))^T K_p\} = n$ in a neighborhood of $\boldsymbol{q}_d$. Now, considering assumption A2 and by recalling the mean-value theorem, we have $\boldsymbol{\xi}_d - \boldsymbol{\xi}(\boldsymbol{q}) = J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}}))[\boldsymbol{q} - \boldsymbol{q}_d]$, with some $\bar{\boldsymbol{q}}$ in the neighborhood of $\boldsymbol{q}_d$. Therefore, we can rewrite $J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p[\boldsymbol{\xi}_d - \boldsymbol{\xi}] = \mathbf{0}$ as

$$J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))^T K_p J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}}))[\boldsymbol{q}_d - \boldsymbol{q}] = \mathbf{0}. \tag{19}$$

On the other hand, notice that, for all $\bar{\boldsymbol{q}}$ in a neighborhood of $\boldsymbol{q}_d$, we have that

$$J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}}))^T K_p J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}})) \tag{20}$$

is a symmetric positive definite matrix because $\operatorname{rank}\{J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}}))\} = n$, K_p is a symmetric nonsingular matrix, and $2p \geq n$ ([35, p. 399]).

Since (20) is a positive-definite matrix and, hence, also nonsingular, then by the continuity established in assumption A3, we can assure that

$$J(\boldsymbol{q}, \boldsymbol{\xi}(\boldsymbol{q}), \boldsymbol{Z}(\boldsymbol{q}))^T K_p J(\bar{\boldsymbol{q}}, \boldsymbol{\xi}(\bar{\boldsymbol{q}}), \boldsymbol{Z}(\bar{\boldsymbol{q}})) \tag{21}$$

is also a square nonsingular matrix for all $\boldsymbol{q}$ and $\bar{\boldsymbol{q}}$ in a neighborhood of $\boldsymbol{q}_d$. This allows us to conclude that $\boldsymbol{q} = \boldsymbol{q}_d$ is the unique solution for (19) in a sufficiently small neighborhood of $\boldsymbol{q}_d$. $\diamond$

As a consequence of *Lemma 1*, it may be concluded that $[\boldsymbol{q}^T \quad \dot{\boldsymbol{q}}^T]^T = [\boldsymbol{q}_d^T \quad \boldsymbol{0}]^T \in \mathbb{R}^{2n}$ is an isolated equilibrium of the closed loop system (18).

In order to carry out the stability analysis, we use the Lyapunov's direct method (see, e.g., [36]). Thus, the following Lyapunov function candidate may be considered:

$$\begin{aligned} &V(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}}) \\ &\quad = \tfrac{1}{2}\dot{\boldsymbol{q}}^T M(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\underbrace{[\boldsymbol{\xi}(\boldsymbol{q}_d) - \boldsymbol{\xi}(\boldsymbol{q})]^T K_p [\boldsymbol{\xi}(\boldsymbol{q}_d) - \boldsymbol{\xi}(\boldsymbol{q})]}_{\tilde{\boldsymbol{\xi}}^T K_p \tilde{\boldsymbol{\xi}}} \end{aligned} \tag{22}$$

which, according to assumption A2, is a locally positive-definite function. The time derivative of the Lyapunov function candidate is given by

$$\dot{V}(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T M(\boldsymbol{q})\ddot{\boldsymbol{q}} + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{M}(\boldsymbol{q})\dot{\boldsymbol{q}} - \tilde{\boldsymbol{\xi}}^T K_p \dot{\boldsymbol{\xi}}(\boldsymbol{q}).$$

By using the closed-loop equation (14) and (18), it follows that

$$\begin{aligned} &\dot{V}(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}}) \\ &\quad = \dot{\boldsymbol{q}}^T \left[J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p [\boldsymbol{\xi}_d - \boldsymbol{\xi}] - K_v \dot{\boldsymbol{q}} - C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \right] \\ &\qquad + \frac{1}{2}\dot{\boldsymbol{q}}^T \dot{M}(\boldsymbol{q})\dot{\boldsymbol{q}} - [\boldsymbol{\xi}_d - \boldsymbol{\xi}]^T K_p J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})\dot{\boldsymbol{q}}. \end{aligned}$$

After some simplifications and using (2), it is finally obtained

$$\dot{V}(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}}) = -\dot{\boldsymbol{q}}^T K_v \dot{\boldsymbol{q}}. \tag{23}$$

Since K_v is by design a positive-definite matrix, hence, $\dot{V}(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}})$ is a globally negative-semidefinite function. Therefore, by invoking the Lyapunov's direct method [36], it can be concluded that $[\boldsymbol{q}^T \quad \dot{\boldsymbol{q}}^T]^T = [\boldsymbol{q}_d^T \quad \boldsymbol{0}]^T \in \mathbb{R}^{2n}$ is a stable equilibrium.

In order to prove asymptotic stability, we take advantage of the autonomous nature of the closed-loop equation (18) which lets us apply the Krasovskii–LaSalle's theorem [36]. In the region

$$\Omega = \left\{ \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} : \quad \dot{V}(\boldsymbol{q}_d - \boldsymbol{q}, \dot{\boldsymbol{q}}) = 0 \right\}$$

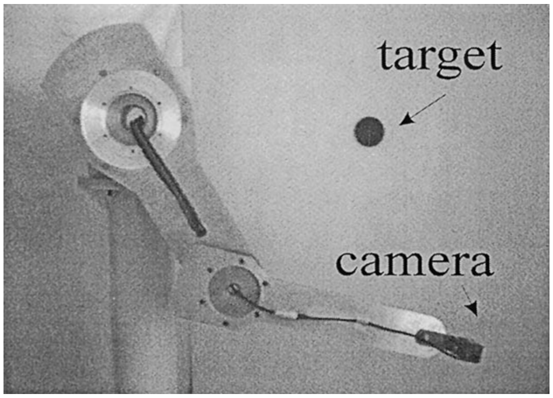


Fig. 4. Experimental arm.

the invariant set is obtained from (17) as $\dot{\boldsymbol{q}} = \boldsymbol{0}$ and $\boldsymbol{q} \in \mathbb{R}^n$: $J(\boldsymbol{q}, \boldsymbol{\xi}, \boldsymbol{Z})^T K_p \tilde{\boldsymbol{\xi}} = \boldsymbol{0}$. Furthermore, according to *Lemma 1*, the latter equation is satisfied for $\boldsymbol{q} = \boldsymbol{q}_d$. Therefore, invoking the Krasovskii–LaSalle's theorem, we can assure that the equilibrium $[\boldsymbol{q}^T \quad \dot{\boldsymbol{q}}^T]^T = [\boldsymbol{q}_d^T \quad \boldsymbol{0}]^T$ is locally asymptotically stable. This means that the control aim is achieved, i.e., $\lim_{t\to\infty} \tilde{\boldsymbol{\xi}}(t) = \boldsymbol{0}$ for a sufficiently small initial feature error $\tilde{\boldsymbol{\xi}}(0)$ and joint velocity $\dot{\boldsymbol{q}}(0)$.

IV. Experimental Case Study

A direct-drive arm with two vertical rigid links has been designed and built at the Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) Research Center, Ensenada, Mexico (see Fig. 4). High-torque brushless direct-drive servos are used to drive the joints without gear reduction. The motors used in the experimental arm are the Compumotor Models DM1200-A and DM1015-B operating in torque mode for the shoulder and elbow joints, respectively. A Panasonic GP-MF502 camera with a lens having a focal length $\lambda = 0.008$ m was attached at the arm tip.

A motion control board based on a TMS320C31 32-bit floating-point microprocessor from Texas Instruments is used to execute the control algorithm. This board is mounted in a PC 486 66-MHz host computer which provides the environment for program generation, compilation, loading data for plotting purposes, and downloading programs for real-time execution [37]. The control program is written in C programming language for the TMS320 compiler and executed in the control board at a 2.5-ms sampling rate. The video signal is acquired by a frame processor Fidelity 200 DT3851-4 from Data Translation mounted in a second PC 486 computer which processes the image and extracts the image features. Data are sent back to the main host computer during robot operation through a serial communication link.

The experiments have shown that Coulomb and viscous friction at the motor joint are present and they depend in a complex manner on joint position and velocity. Instead of modeling these friction phenomena for compensation purposes, they were considered as unmodeled dynamics.

The complete robot dynamics (arm plus camera) has the structure (1). With reference to the symbols listed in Table I, we present below the entries of robot dynamics.

TABLE I
PARAMETERS OF THE MANIPULATOR

	notation	*value*	*unit*
Length link 1	l_1	0.45	m
Length link 2	l_2	0.55	m
Link (1) center of gravity	l_{c1}	0.091	m
Link (2) center of gravity	l_{c2}	0.105	m
Mass link 1	m_1	23.90	kg
Mass link 2 (camera included)	m_2	4.44	kg
Inertia link 1	I_1	1.27	kg m^2
Inertia link 2 (camera included)	I_2	0.24	kg m^2
Grayity acceleration	g	9.8	m/sec^2

TABLE II
PARAMETERS OF THE CAMERA

	notation	*value*	*unit*
Focal length	λ	0.008	m
Scaling factor	α	72,727	pixels/m

The elements $M_{ij}(\boldsymbol{q})$ $(i, j = 1, 2)$ of the inertia matrix $M(\boldsymbol{q})$ are

$$\begin{aligned} M_{11}(\boldsymbol{q}) &= m_1 l_{c1}^2 + m_2\left(l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos(q_2)\right) + I_1 + I_2 \\ M_{12}(\boldsymbol{q}) &= m_2\left(l_{c2}^2 + l_1 l_{c2}\cos(q_2)\right) + I_2 \\ M_{21}(\boldsymbol{q}) &= m_2\left(l_{c2}^2 + l_1 l_{c2}\cos(q_2)\right) + I_2 \\ M_{22}(\boldsymbol{q}) &= m_2 l_{c2}^2 + I_2. \end{aligned}$$

The elements $C_{ij}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ $(i, j = 1, 2)$ from the centrifugal and Coriolis matrix $C(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are

$$\begin{aligned} C_{11}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= -m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2 \\ C_{12}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= -m_2 l_1 l_{c2}\sin(q_2)\left(\dot{q}_1 + \dot{q}_2\right) \\ C_{21}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= m_2 l_1 l_{c2}\sin(q_2)\dot{q}_1 \\ C_{22}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= 0. \end{aligned}$$

The entries of the gravitational torque vector $\boldsymbol{g}(\boldsymbol{q})$ are given by

$$\begin{aligned} g_1(\boldsymbol{q}) &= (m_1 l_{c1} + m_2 l_1)g\sin(q_1) + m_2 l_{c2} g\sin(q_1 + q_2) \\ g_2(\boldsymbol{q}) &= m_2 l_{c2} g\sin(q_1 + q_2). \end{aligned}$$

Among the entries of the robot dynamics, only the gravitational torque vector $\boldsymbol{g}(\boldsymbol{q})$ is needed for implementing the visual servo controller (16).

The image provided by the frame processor has a 640 × 480 pixels resolution. From the technical specifications given by the camera manufacturer, we found two slightly different scale factors. However, for the sake of simplicity, we have used an average value as the scale factor α. Table II lists the numerical values of the camera model (5).

The geometric description of the robotic system is sketched in Fig. 5. The robot frame is described by its main axes x_r, y_r, and z_r where y_r stands for the vertical axis. The camera frame denoted by the axes x_c, y_c, and z_c is located at the robot tip (end-effector frame). The axes z_r and z_c are assumed to be parallel and the plane x_c–y_c is assumed to be laying on the x_r–y_r plane. For $q_1 = q_2 = 0$, the camera frame has π rad rotation around the axis z_r with respect to the robot frame. By denoting with ϕ, ϑ, and ψ, the ZYZ Euler angles [24] for parameterization of camera orientation, we obtain the direct kinematics equation

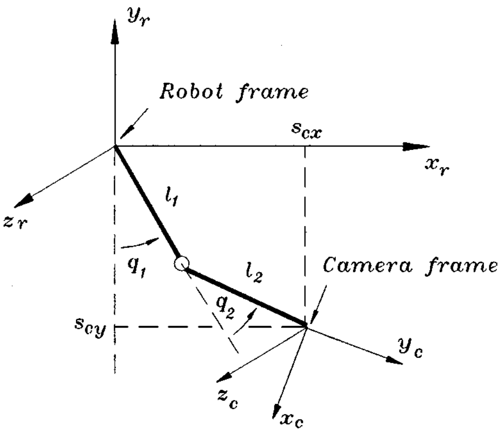


Fig. 5. Coordinate frames.

$$\begin{bmatrix} s_{cx} \\ s_{cy} \\ s_{cz} \\ \phi \\ \vartheta \\ \psi \end{bmatrix} = \begin{bmatrix} l_1\sin(q_1) + l_2\sin(q_1 + q_2) \\ -l_1\cos(q_1) - l_2\cos(q_1 + q_2) \\ 0 \\ q_1 + q_2 + \pi \\ 0 \\ 0 \end{bmatrix}$$

where we have used the fact that $\phi = q_1 + q_2 + \pi$ and $\vartheta = \psi = 0$, because the arm moves in the plane x_r–y_r.

The analytical Jacobian is computed by differentiating the direct kinematics

$$J_A(\boldsymbol{q}) = \begin{bmatrix} l_1\cos(q_1) + l_2\cos(q_1 + q_2) & l_2\cos(q_1 + q_2) \\ l_1\sin(q_1) + l_2\sin(q_1 + q_2) & l_2\sin(q_1 + q_2) \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{24}$$

Considering the Euler angles ZYZ, the transformation matrix $T(\boldsymbol{q})$ which relates the camera angular velocity to the time derivative of the Euler angles is [24]

$$T(\boldsymbol{q}) = \begin{bmatrix} 0 & -\sin(q_1 + q_2 + \pi) & 0 \\ 0 & \cos(q_1 + q_2 + \pi) & 0 \\ 1 & 0 & 1 \end{bmatrix}. \tag{25}$$

Finally, the rotation matrix of the camera frame with respect to the robot frame is given by

$$R_c(\boldsymbol{q})^T = \begin{bmatrix} \cos(q_1 + q_2 + \pi) & -\sin(q_1 + q_2 + \pi) & 0 \\ \sin(q_1 + q_2 + \pi) & \cos(q_1 + q_2 + \pi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{26}$$

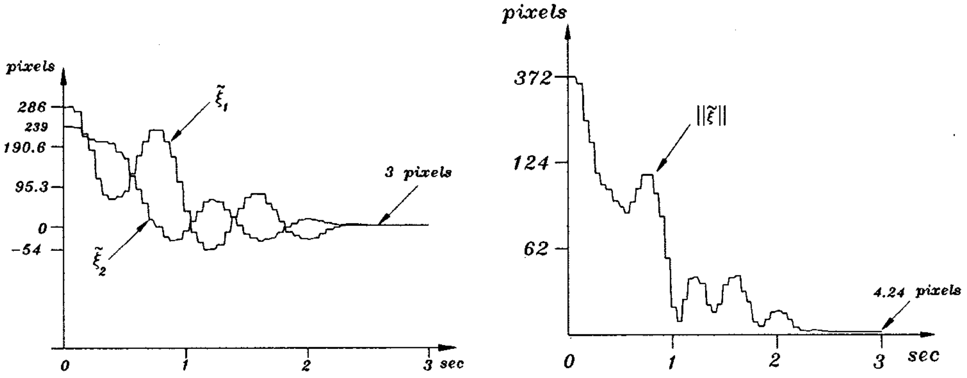


Fig. 6. Feature position errors and norm.

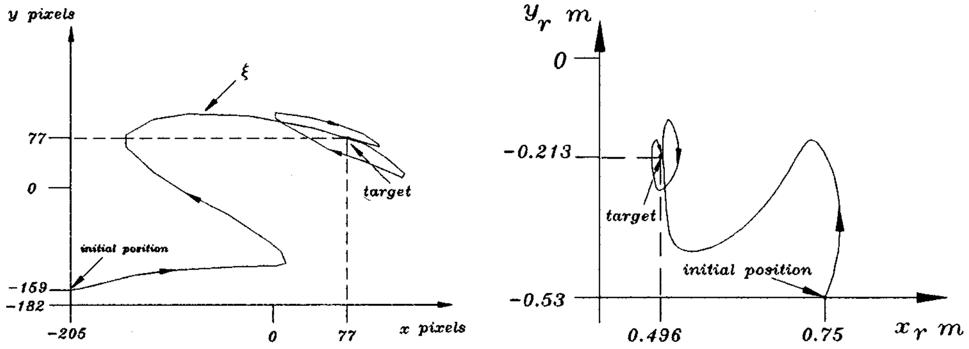


Fig. 7. Feature position trajectory in image plane and path of the camera in the robot workplane.

It is worth mentioning that several physical phenomena have been neglected in the description of the experimental setup, such as friction at the joints, lens radial distortion, and possible camera misalignment.

In the following section, we describe the results obtained from experimental tests using one and two feature points. In both cases, a whiteboard was located at a distance $Z = -0.74$ m in front of the camera and parallel to the plane where the robot moves. For the former, a fixed black disc was placed on the board, while, in the latter, two discs were considered. The centroid of each disc was selected as the object feature point.

The two experimental tests were carried out with the following initial configuration for the robot: $q_1(0) = 30°$, $q_2(0) = 45°$, and $\dot{q}_1(0) = \dot{q}_2(0) = 0$.

A. One Feature Point

In the first experimental test, we have considered only one object feature point. According to (15), by using the image Jacobian (11) together with (24)–(26) and after some manipulation, the following Jacobian is obtained:

$$J(\boldsymbol{q}, \boldsymbol{\xi}, Z) = \begin{bmatrix} -\frac{\alpha\lambda}{Z} l_1 \cos(q_2) - \frac{\alpha\lambda}{Z} l_2 + y & -\frac{\alpha\lambda}{Z} l_2 + y \\ \frac{\alpha\lambda}{Z} l_1 \sin(q_2) - x & -x \end{bmatrix}.$$

The proposed controller given by (16) was used to control the robot. The controller parameters were selected as

$$K_p = \begin{bmatrix} 2 \times 10^{-4} & 0 \\ 0 & 6 \times 10^{-4} \end{bmatrix} \quad \mathrm{N \cdot m/pixels^2}$$

and

$$K_v = \begin{bmatrix} 0.10 & 0 \\ 0 & 0.22 \end{bmatrix} \quad \mathrm{N \cdot m \cdot s/deg.}$$

At the initial configuration of the robot, the image feature point was $\boldsymbol{\xi}(0) = [-205 \quad -159]^T$ pixels. The desired image feature point was set to $\boldsymbol{\xi}_d = [80 \quad 80]^T$ pixels. With these data, the initial image feature error was $\tilde{\boldsymbol{\xi}}(0) = [285 \quad 239]^T$ pixels.

The visual servo controller was implemented with two sampling rates. Velocity and gravity compensation feedbacks were updated every 2.5 ms, while the visual feedback loop every 50 ms.

The experimental results are shown in Figs. 6 and 7. Fig. 6 depicts the time evolution of the elements of the feature error vector $\tilde{\boldsymbol{\xi}}$. After the transient, both components tend asymptotically to a small neighborhood of zero (3 pixels). They remain stuck close to zero due to the presence of static friction at the joints. This real-world phenomenon is not predicted by the friction-free theoretical analysis which states that the feature error vector must vanish as time increases. Unless a precise friction model (including stiction) is incorporated in the analysis, the computation of the convergence time and the steady-state pixels errors cannot be accurately obtained for direct visual servo controllers.

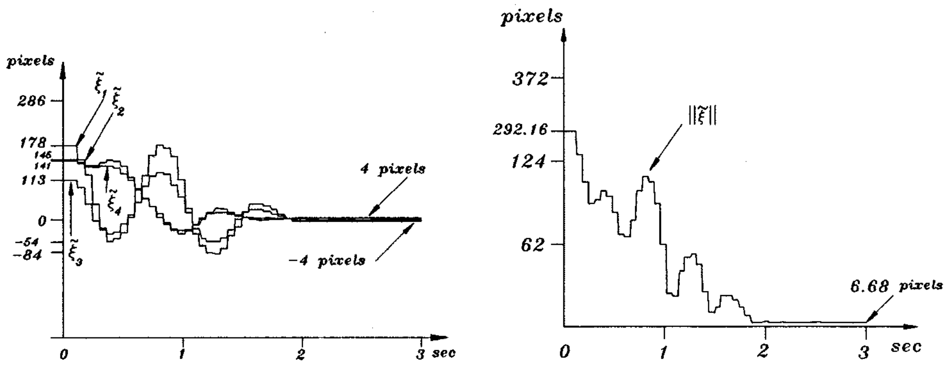


Fig. 8. Image feature errors and norm.

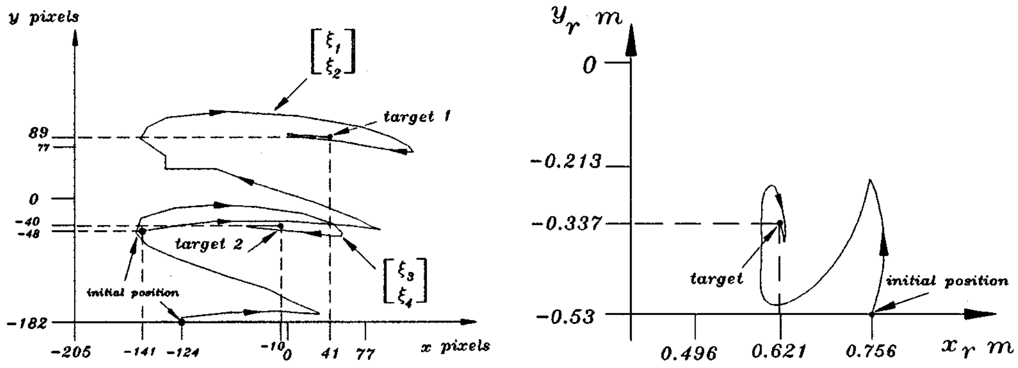


Fig. 9. Image features trajectory in image plane and path of the camera in the robot workplane.

Fig. 6 also depicts the evolution of the norm of the image feature error; the initial value at the beginning was $\|\tilde{\boldsymbol{\xi}}(0)\| = 372$ pixels and at $t = 3$ s the value was $\|\tilde{\boldsymbol{\xi}}(3)\| \approx 5$ pixels.

It is also interesting to describe the shape of the image feature point viewed in the image frame. The trajectory of the feature $\boldsymbol{\xi}$ in the image plane is presented in Fig. 7, which shows the convergence to a small neighborhood of the desired feature point $\boldsymbol{\xi}_d = [80 \quad 80]$ pixels. The oscillatory behavior may be due mainly to the friction at the joints, and possible lens radial distortions.

Finally, Fig. 7 also illustrates the trajectory of the origin of the camera frame on the x_r–y_r plane. The distance between initial and destination positions was 0.41 m.

B. Two Feature Points

In the second experimental test, two black discs were placed on the board ($p = 2$). The disc centroids were considered as the object feature points, while the image feature vector $\boldsymbol{\xi} = [x_1 \; y_1 \; x_2 \; y_2]^T$ represents their projections in the image plane. The Jacobian matrix is obtained from its definition (15), the extended image Jacobian (13), and (24)–(26). This leads to

$$
\begin{aligned}
&J(\boldsymbol{q}, \boldsymbol{\xi}, Z)\\
&= \begin{bmatrix}
-\dfrac{\alpha\lambda}{Z} l_1 \cos(q_2) - \dfrac{\alpha\lambda}{Z} l_2 + y_1 & -\dfrac{\alpha\lambda}{Z} l_2 + y_1 \\
\dfrac{\alpha\lambda}{Z} l_1 \sin(q_2) - x_1 & -x_1 \\
-\dfrac{\alpha\lambda}{Z} l_1 \cos(q_2) - \dfrac{\alpha\lambda}{Z} l_2 + y_2 & -\dfrac{\alpha\lambda}{Z} l_2 + y_2 \\
\dfrac{\alpha\lambda}{Z} l_1 \sin(q_2) - x_2 & -x_2
\end{bmatrix}.
\end{aligned}
$$

The parameters of controller (16) were selected as

$$
K_p = \begin{bmatrix}
7 \times 10^{-4} & 0 & 0 & 0 \\
0 & 2 \times 10^{-4} & 0 & 0 \\
0 & 0 & 7 \times 10^{-4} & 0 \\
0 & 0 & 0 & 2 \times 10^{-4}
\end{bmatrix}
\; \mathrm{N \cdot m/pixels^2}
$$

$$
K_v = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.07 \end{bmatrix} \; \mathrm{N \cdot ms/deg.}
$$

At the initial configuration of the robot, the image feature point was $\boldsymbol{\xi}(0) = [-141 \quad -48 \quad -124 \quad -182]^T$ pixels. The desired image feature point was set to $\xi_d = [37 \;\; 93 \;\; -11 \;\; -37]^T$ [pixels]. With these data, the initial image feature error was $\tilde{\xi}(0) = [178 \;\; 141 \;\; 113 \;\; 145]^T$ pixels.

As in the previous experiment, the visual servo controller was implemented with two sampling rates. The velocity and gravity compensation feedback signals were updated every 2.5 ms, while the visual feedback loop every 55 ms.

Figs. 8 and 9 depict the experimental results. The components of the image feature error are shown in Fig. 8. The four components present a decreasing tendency and, actually, they remain within the neighborhood of 5 pixels around zero after 2 s. Also, Fig. 8 depicts the evolution of the norm of the image feature error which starts at 292 pixels and ends at about 7 pixels.

The path of the image feature vector in the image plane is shown in Fig. 9. The two curves correspond to the path of each centroid starting at the initial configuration $\xi(0)$ and ending in a neighborhood of the desired image feature ξ_d.

The path of the camera center (end effector) in plane x_r–y_r is shown in Fig. 9. In the robot workspace, the distance from the initial camera position to the destination was approximately 0.24 m.

V. Conclusions

In this paper, we have presented an image-based direct visual servo controller for camera-in-hand robot manipulators. The controller is of a simple structure based on a transpose Jacobian term plus gravity compensation which feeds back directly the image feature errors and the joint velocities. This controller is capable of placing the camera mounted on the robot in a desired relative position with respect to a static object. By using the Lyapunov's direct method, we have shown the stability of the closed-loop system. It should be emphasized that the full nonlinear robot dynamics has been included in the analysis. Local asymptotic stability can be achieved under weak assumptions on the Jacobian by invoking the Krasovskii–LaSalle's theorem. Experimental results with a two-degree-of-freedom arm, using one and two feature points, were presented to illustrate the control system stability and performance.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 651–670, Oct. 1996.

[2] B. Hove and J. J. E. Slotine, "Experiments in robotic catching," in *Proc. American Control Conf.*, Boston, MA, June 1991, pp. 380–385.

[3] S. W. Wijesoma, D. F. H. Wolfe, and R. J. Richards, "Eye-to-hand coordination for vision-guided robot control applications," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 63–78, Feb. 1993.

[4] P. K. Allen, A. Tomcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Automat.*, vol. 9, pp. 152–165, Apr. 1993.

[5] M. Lei and B. K. Ghosh, "Visually guided robotic tracking and grasping of a moving object," in *Proc. 32nd Conf. Decision and Control*, San Antonio, TX, Dec. 1993, pp. 1604–1609.

[6] G. D. Hager, W. C. Chang, and A. S. Morse, "Robot hand-eye coordination based on stereo vision," *IEEE Contr. Syst. Mag.*, vol. 15, pp. 30–39, Feb. 1995.

[7] G. D. Hager, "A modular system for robust positioning using feedback from stereo vision," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 582–595, Aug. 1997.

[8] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 759–766, Oct. 1996.

[9] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Automat.*, vol. 3, pp. 404–417, Sept. 1987.

[10] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servoing," in *Proc. IEEE Int. Conf. Robotics and Automation*, Sacramento, CA, 1991, pp. 2267–2272.

[11] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Weighted selection of image features for resolved rate visual feedback control," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 31–47, Feb. 1991.

[12] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 313–326, June 1992.

[13] H. Hashimoto, T. Kubota, M. Sato, and F. Harashima, "Visual control of robotic manipulator based on neural networks," *IEEE Trans. Ind. Electron.*, vol. 39, pp. 490–496, Dec. 1992.

[14] F. Chaumette, P. Rives, and B. Espiau, "Classification and realization of the different vision-based tasks," in *Visual Servoing*, K. Hashimoto, Ed, Singapore: World Scientific, 1993, pp. 199–228.

[15] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision," *IEEE Trans. Robot. Automat.*, vol. 9, Feb. 1993.

[16] K. Hashimoto and H. Kimura, "Dynamic visual servoing with nonlinear model-based control," in *Proc. 1993 IFAC World Congr.*, vol. 9, Sydney, Australia, July 1993, pp. 405–408.

[17] P. I. Corke, "Video-rate robot visual servoing," in *Visual Servoing*, K. Hashimoto, Ed, Singapore: World Scientific, 1993, pp. 257–283.

[18] A. Castaño and S. Hutchinson, "Visual compliance: Task-directed visual servo control," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 334–342, June 1994.

[19] M. Takegaki and S. Arimoto, "A new feedback method for dynamic control of manipulators," *Trans. ASME, J. Dynam. Syst., Meas., Contr.*, vol. 103, pp. 119–125, June 1981.

[20] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "A stable visual servo controller for camera-in-hand robotic systems," in *Proc. 8th Int. Conf. Advanced Robotics (ICAR)*, Monterey, CA, July 1997, pp. 117–123.

[21] A. Maruyama and M. Fujita, "Robust visual servo control for planar manipulators with the eye-in-hand configuration," in *Proc. IEEE Int. Conf. Decision and Control*, vol. 3, San Diego, CA, Dec. 1997, pp. 2551–2552.

[22] M. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.

[23] D. Koditschek, "Natural motion for robot arms," in *Proc. 23rd Conf. Decision and Control*, Las Vegas, NV, Dec. 1984, pp. 733–735.

[24] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. New York: McGraw-Hill, 1996.

[25] K. Hashimoto, A. Aoki, and T. Noritsugu, "Visual servoing with redundant features," in *Proc. 35th Conf. Decision and Control*, Kobe, Japan, Dec. 1996, pp. 2482–2483.

[26] D. E. Koditschek, "Natural control of robot arms," Center for Systems Science, Yale University, New Haven, CT, revised Feb. 1985, Dec. 1984.

[27] F. Miyazaki and S. Arimoto, "Sensory feedback for robot manipulators," *J. Robot. Syst.*, vol. 2, no. 1, pp. 53–71, 1985.

[28] S. Arimoto, F. Miyazaki, H. G. Lee, and S. Kawamura, "Rivival of Lyapunov's direct method in robot control and design," in *Proc. American Control Conf.*, Atlanta, GA, June 1988, pp. 1764–1769.

[29] J. J. Craig, *Introduction to Robotics: Mechanics and Control*, 2nd ed. Reading, MA: Addison-Wesley, 1989.

[30] F. Miyazaki and Y. Masutani, "Robustness of sensory feedback control based on imperfect Jacobian," in *Robotics Research: The Fifth International Symposium*, H. Miura and S. Arimoto, Eds. Cambridge, MA: MIT Press, 1990, pp. 4201–208.

[31] D. E. Koditschek, "Some applications of natural motion control," *Trans. ASME, J. Dynam. Syst., Meas., Contr.*, vol. 113, pp. 552–557, Dec. 1991.

[32] R. Kelly, "Regulation of manipulators in generic task space: An energy shaping plus damping injection approach," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 381–386, Apr. 1999.

[33] R. Kelly and A. Coello, "Analysis and experimentation of transpose Jacobian-based Cartesian regulators," *Robotica*, vol. 17, no. 3, pp. 303–312, May/June 1999.

[34] C. C. Cheah, S. Kawamura, and S. Arimoto, "Feedback control for robotic manipulators with an uncertain Jacobian matrix," *J. Robot. Syst.*, vol. 16, no. 2, pp. 119–134, 1999.

[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[36] M. Vidyasagar, *Nonlinear Systems Analysis*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[37] F. Reyes and R. Kelly, “A direct-drive robot for control research,” in *Proc. IASTED Int. Conf.: Applications of Control and Robotics*, Orlando, FL, Jan. 1996, pp. 181–184.

Rafael Kelly (S’84–M’89) was born in Monterrey, Mexico, in 1959. He received the B.S. degree in physics from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, Mexico, and the Ph.D. degree in automatic control from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1980 and 1986, respectively.

He is currently a Professor at the Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Mexico. His research interests include adaptive control systems, robot control, vision systems, and neural networks.

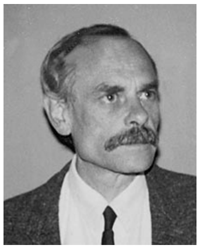
Ricardo O. Carelli (M’76–SM’98) was born in San Juan, Argentina. He received the Electrical Engineering degree (with honors) from the National University of San Juan, San Juan, Argentina, and the Ph.D. degree in electrical engineering from the National University of Mexico, Mexico City, Mexico, in 1976 and 1989, respectively.

He is presently a Full Professor at the National University of San Juan and a Senior Research Worker of the National Council for Scientific and Technical Research (CONICET, Argentina). Since 1990, he has been the Adjunct Director of the Instituto de Automática, National University of San Juan. His research interests are robotics, manufacturing systems, adaptive control, and artificial intelligence applied to automatic control.

Prof. Carelli is a senior member of AADECA-IFAC.

Oscar Nasisi was born in San Luis, Argentina, in 1961. He received the Electronics Engineering degree from the National University of San Juan, San Juan, Argentina, the M.S. degree in electronics engineering from the National Universities Foundation for International Cooperation, Eindhoven, The Netherlands, and the Ph.D. degree from the National University of San Juan in 1986, 1989, and 1998, respectively.

Since 1986, he has been with the Instituto de Automática, National University of San Juan, where he currently is a Full Professor. His research areas of interest are artificial vision, robotics, and adaptive control.



Benjamín Kuchen (M’90) was born in Santa María, Argentina. He received the Electrical Engineering degree from the Catholic University of Cordoba, Cordoba, Argentina, and the Ph.D. degree in engineering from RWTH-Aachen, Aachen, Germany, in 1967 and 1974, respectively.

He is presently a Full Professor at the National University of San Juan San Juan, Argentina, and a Senior Research Worker of the National Council for Scientific and Technical Research (CONICET, Argentina). Since 1983, he has been the Director of the Instituto de Automática, National University of San Juan. His research interests are in robotics and discrete-events systems.

Prof. Kuchen is a member of AADECA-IFAC.

Fernando Reyes was born in Puebla, Mexico, in 1962. He received the M.S. degree in electronics from the Instituto Nacional de Astrofísica Optica y Electrónica, Puebla, Mexico, and the Ph.D. degree from the Centro de Investigación Científica y de Educación Superior de Ensenada, Ensenada, Mexico, in 1989 and 1997, respectively.

He is currently a Professor at the Universidad Autónoma de Puebla, Puebla, Mexico. His research interests are robotics and automatic control.