We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS BOOK CITATION INDEX INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

20

Vision-based 2D and 3D Control of Robot Manipulators

Luis Hernández[1], Hichem Sahli[2] and René González[3]
[1]*Universidad Central de Las Villas (UCLV),* [2]*Vrije Universiteit Brussel (VUB),*
[3]*Empresa de Automatización Integral (CEDAI)*
[1]*Cuba,* [2]*Belgium,* [3]*Cuba*

1. Introduction

Robotics has been a paradigm for science in the last few decades. At first, scientists' efforts were devoted to the solution of the problem of planning and control of the motion of robot manipulators. However, the motion control of robot manipulators in unstructured environments is today an attractive scientific problem. An interesting solution for motion control is the use of sensor information, such as computer vision, in the system's feedback. Several works and tools have been developed in recent years in this field (Corke; 2005), (Chaumette and Hutchinson; 2006). The more typical approaches consider visual perception for servoing and for the so called *look and move* (Hutchinson et al.; 1996). Visual servoing (Kelly et al.; 2000) can be classified into two approaches: camera-in-hand or camera-to-hand (Flandin et al.; 2000). In camera-to-hand robotic systems, multiple cameras or a single camera fixed in the world-coordinate frame capture images of both, the robot and its environment. The tracking of the object with visual feedback can be made in 2D or 3D. An interesting solution for the visual servoing of camera-in hand robot manipulators in 2D can be found in e. g. (Bonfe et al.; 2002) and (Hernández et al.; 2008) where stability demonstration of a decoupled controller have been presented. For 3D tracking some solutions reported are *look and move* controller, with one camera (Sim et al.; 2002) or more than one camera (Xie et al.; 2005). A 3D visual servoing with stability analysis in continuous time is presented by (Hernández et al.; 2008a) and (Kelly et al.; 2006) present a direct visual servoing with transpose Jacobian control technique for regulation of robot manipulators in the 3D Cartesian space. In a similar vein (Enescu et al.; 2006) present mobile robot navigation for person tracking using a stereo head-camera.
In this chapter, we consider the control problem of camera-in-hand robot manipulators in 2D and 3D. In both cases only one camera is mounted on the robot's arm, which supplies visual information of the environment, with the aim of moving the manipulator by maintaining the image of the tracked object (a sphere) in the centre of the image plane, despite the possible movements of the object. In the 3D control the constant radius is used as a feature too. In this work, the proposed control system considers two loops in cascade, an internal loop solving the robots' joint control, and an external loop implementing a dynamic look and move visual controller. A stability analysis in discrete time is developed under the conditions that it is possible to approximate the dynamic effect of the internal loop as an external loop time delay (Corke; 1996) and (Bonfe et al.; 2002). The more classic presentations of servovisual control are velocity controllers, based in the term *feature Jacobian*,(Chaumette and Hutchinson; 2006)

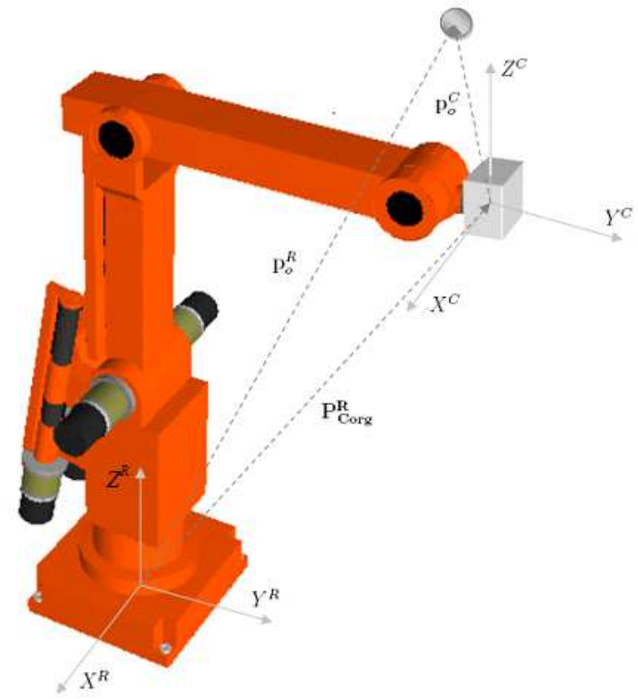


Fig. 1. Camera-in-hand robotic system

(Chaumette and Hutchinson; 2007), but we present the robotic and vision systems modeled for small variations about the operating point for position control and in these conditions the stability of the whole system are balanced.
A particular study is made using an ASEA IRB6 robot manipulator which has mechanically decoupled wrist. This allows keeping the orientation of the camera's optical axis while the arm is moving.
In order to validate the proposed control system, a general stability analysis is presented and an analysis of the step response in a regulator type system is made. The disturbance is interpreted as initial conditions. To illustrate the proposed controller, the control system stability and its performance, both simulation results as well as experimental results using the ASEA IRB6 robot manipulator are presented. Our experimental results confirm the expected step response in the image plane, with good time performance and zero steady-state error.

2. Robotic System Model

As shown in Fig. 1, the robotic system considered has a robot manipulator with a camera in its hand. The basic mathematical description of this system consists of the robot and the camera model.

2.1 Robot Kinematics Model

The kinematics of a manipulator gives the relationship between the joint positions $\mathbf{q}$ and the corresponding tool translational (x, y, z) and angular position (α, β, γ). For an n-axis rigid link manipulator, the forward kinematic solution, $\mathbf{T}$, could be computed for any manipulator,

irrespective of the number of joints or kinematic structure (Barrientos et al.; 1997). A generic mathematical representation could be:

$$\left[\begin{array}{cccccc} x & y & z & \alpha & \beta & \gamma \end{array} \right]^T = f(q_1, q_2, \ldots, q_n) = \mathbf{T} \tag{1}$$

For manipulator path planning, the inverse kinematic solution $\mathbf{T^{-1}}$ gives the joint angles $\mathbf{q}$ required to reach the specified tool's position. In general this solution is non-unique (Barrientos et al.; 1997). A generic mathematical representation could be:

$$\mathbf{q} = \left[\begin{array}{cccc} q_1 & q_2 & \ldots & q_n \end{array} \right]^T = g(x, y, z, \alpha, \beta, \gamma) = \mathbf{T}^{-1} \tag{2}$$

2.2 Robot Dynamics Model

In the absence of friction or other disturbances, the dynamics of a serial n-link rigid robot manipulator can be written as (Kelly and Santibáñez; 2003)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau \tag{3}$$

where:

$\mathbf{M}(\mathbf{q})$ $n \times n$ symmetric positive definite manipulator inertia matrix
$\mathbf{q}$ $n \times 1$ vector of joint displacements
$\mathbf{C}(\mathbf{q})$ $n \times 1$ vector of centripetal and Coriolis torques
$\mathbf{g}(\mathbf{q})$ $n \times 1$ vector of gravitational torques
τ $n \times 1$ vector of applied joint torques

2.3 Camera Model

According to Fig. 1 we consider a vision system mounted on the robot tool, with coordinate frame, Σ_C, which moves in the space $< X^R, Y^R, Z^R >$ of the robot coordinate frame Σ_R. The origin of the camera coordinate frame (tool frame) with respect to the robot coordinate frame is represented by the vector $\mathbf{P^R_{Corg}}$ with coordinates $\left[\begin{array}{ccc} p^R_{xc} & p^R_{yc} & p^R_{zc} \end{array} \right]^T \in \Re^3$.

Points of interest in the workspace are identified as $\mathbf{p}^C_o$ with coordinates $\left[\begin{array}{ccc} p^C_{xo} & p^C_{yo} & p^C_{zo} \end{array} \right]^T \in \Re^3$ in the camera reference system Σ_C, and $\mathbf{p}^R_o = \left[\begin{array}{ccc} p^R_{xo} & p^R_{yo} & p^R_{zo} \end{array} \right]^T \in \Re^3$ in the robot reference system Σ_R.

2.3.1 Camera Model for Visual-Based 2D Control

The image acquired by the camera supplies a two-dimensional (2D) array of brightness values from a three-dimensional (3D) scene. This image may undergo various types of computer processing to enhance image properties and extract image features. In this paper we work with a known spherical object with radius r_o. The centre of gravity is used as object features (state) for the 2D control. We assume that the image features are the projection into the 2D image plane of the 3D characteristics of the scene.

The object moves in a plane parallel to the plane $< Y^C, Z^C >$ of the camera coordinate frame Σ_C. The camera's optical axis coincides with X^C axis.

A perspective projection with a focal λ is assumed. The point $\mathbf{p}^C_o$ with coordinates $\left[\begin{array}{ccc} p^C_{xo} & p^C_{yo} & p^C_{zo} \end{array} \right]^T$ in the camera frame projects onto a point (u, v) (pixel) on the image plane.

Let $\mathbf{p}_o^C$ be the position of the object's centre of gravity. According to the perspective projection (Hutchinson et al.; 1996), we have

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -\alpha \frac{\lambda}{p_{xo}^C} \begin{bmatrix} p_{yo}^C \\ p_{zo}^C \end{bmatrix} \tag{4}$$

where α is the scaling factor in pixels per meter due to the camera sampling, λ is the focal length of the camera lens. This model is also called the imaging model (Kelly et al.; 2000).
The object distance p_{xo}^C along the camera's optical axis X^C is constant.
The orientation of the camera frame with respect to the robot frame in the plane $< Y^R, Z^R >$ of Σ_R is denoted by $\mathbf{R}_R^C = \mathbf{R}_R^C(\psi) \in SO(2)$
where

$$\mathbf{R}_R^C(\psi) = \begin{bmatrix} C\psi & -S\psi \\ S\psi & C\psi \end{bmatrix} \tag{5}$$

Where Si is $sin(i)$ and Ci is $cos(i)$, with ψ being the mechanical angle between Σ_C and Σ_R. Note that $\mathbf{R}_R^C(\psi)$ is an orthogonal matrix.
Following the configuration of Fig. 1, and taking into account equation (5), it is possible to obtain the y and z components of vector $\mathbf{p}_o^C$ as:

$$\begin{bmatrix} p_{yo}^C \\ p_{zo}^C \end{bmatrix} = \mathbf{R}_R^C(\psi)^T \left(\begin{bmatrix} p_{yo}^R \\ p_{zo}^R \end{bmatrix} - \begin{bmatrix} p_{yc}^R \\ p_{zc}^R \end{bmatrix} \right)$$

and finally, according to (4) and (5)we obtain:

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -h \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} \left(\begin{bmatrix} p_{yo}^R \\ p_{zo}^R \end{bmatrix} - \begin{bmatrix} p_{yc}^R \\ p_{zc}^R \end{bmatrix} \right) \tag{6}$$

Where $h = \alpha \frac{\lambda}{p_{xo}^C}$.

2.3.2 Camera Model for Visual-Based 3D Control

In the 3D control the object moves in the space $< X^R, Y^R, Z^R >$ of the robot coordinate frame Σ_R; and the camera's optical axis coincide with the Z^C axis of the camera's coordinate frame Σ_C. We continue work with a known spherical object with radius r_o; and in this case the object features (state) are the centre of gravity and image radio.
According to this consideration Equation (4) becomes:

$$\xi = \begin{bmatrix} u \\ v \end{bmatrix} = -\alpha \frac{\lambda}{p_{zo}^C} \begin{bmatrix} p_{xo}^C \\ p_{yo}^C \end{bmatrix} \tag{7}$$

In order to estimate the object's distance p_{zo}^C along the camera's optical axis Z^C it is possible to use a well-known object size and the corresponding apparent size in the image plane (Corke; 1996). Our object is a sphere with radius r_o and the apparent image radius is r. Following Equation (7), p_{zo}^C, r_o and r can be related by,

$$r = -\frac{\alpha \lambda r_o}{p_{zo}^C} \tag{8}$$

Finally combining Equations (7) and (8) we define the object state (feature) vector:

$$\xi' = \begin{bmatrix} u \\ v \\ r \end{bmatrix} = -\alpha \frac{\lambda}{p_{zo}^C} \begin{bmatrix} p_{xo}^C \\ p_{yo}^C \\ r_o \end{bmatrix} \tag{9}$$

The orientation of the camera frame, Σ_C, with respect to the robot frame, Σ_R is denoted by $\mathbf{R'}_C^R = \mathbf{R'}_C^R(\phi, \theta, \psi) \in SO(3)$ where $\mathbf{R}_C^R$ can be described by Euler angles (Barrientos et al.; 1997).
Let (ϕ, θ, ψ) being the given set of Euler angles, the following rotations: Frame rotation by the angle ϕ about axis Z, frame rotation by the angle θ about axis Y' and frame rotation by the angle ψ about axis Z''.
In this case the rotation matrix is:

$$\mathbf{R'}_C^R(\phi, \theta, \psi) = \begin{bmatrix} C\phi C\theta C\psi - S\phi S\psi & -C\phi C\theta S\psi - S\phi C\psi & C\phi S\theta \\ S\phi C\theta C\psi + C\phi S\psi & -S\phi C\theta S\psi + C\phi C\psi & S\phi S\theta \\ -S\theta C\psi & S\theta S\psi & C\theta \end{bmatrix} \tag{10}$$

Following the configuration of Fig. 1, and taking into account Equation (10), it is possible to obtain the components of vector $\mathbf{p}_o^C$ as:

$$\begin{bmatrix} p_{xo}^C \\ p_{yo}^C \\ p_{zo}^C \end{bmatrix} = \mathbf{R'}_C^R(\phi, \theta, \psi)^T \left(\begin{bmatrix} p_{xo}^R \\ p_{yo}^R \\ p_{zo}^R \end{bmatrix} - \begin{bmatrix} p_{xc}^R \\ p_{yc}^R \\ p_{zc}^R \end{bmatrix} \right) \tag{11}$$

2.3.3 Linear Camera Model

According to Fig. 1, the axes Z^R and Z^C are parallel and, for simplicity in the analysis, they are taken with the same direction. In this case, the Euler's angles are $\phi = 0$, $\theta = 0$ and ψ varies according the rotation of the robot's base. In these conditions Equation (10) becomes:

$$\mathbf{R'}_C^R(\psi) = \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

According to the assumption $A2.1$, only a small variation about the operating point will be taken into account. If we set $\psi_0 = 0$ as operating point and take the linear approximation of (12), the variations of Equation (11) can be written as:

$$\begin{bmatrix} \delta p_{xo}^C \\ \delta p_{yo}^C \\ \delta p_{zo}^C \end{bmatrix} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \end{bmatrix} \left(\begin{bmatrix} \delta p_{xo}^R \\ \delta p_{yo}^R \\ \delta p_{zo}^R \end{bmatrix} - \begin{bmatrix} \delta p_{xc}^R \\ \delta p_{yc}^R \\ \delta p_{zc}^R \end{bmatrix} \right) \tag{13}$$

We can express their perturbation using Taylor series expansion. Since we look for a linear dependency on the variables, we only use the first order terms of the series. With this analysis Equation (9) becomes:

$$\delta\xi' = \begin{bmatrix} \delta u \\ \delta v \\ \delta r \end{bmatrix} = -\alpha \frac{\lambda}{p_{zo}^C} \begin{bmatrix} \delta p_{xo}^C \\ \delta p_{yo}^C \\ -\frac{r_o \delta p_{zo}^C}{p_{zo}^C} \end{bmatrix} \tag{14}$$

and finally, according to (13) and (14) we obtain:

$$\delta\xi' = -\alpha\frac{\lambda}{p^C_{zo}}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{r_o}{p^C_{zo}} \end{bmatrix}\left(\begin{bmatrix} \delta p^R_{xo} \\ \delta p^R_{yo} \\ \delta p^R_{zo} \end{bmatrix} - \begin{bmatrix} \delta p^R_{xc} \\ \delta p^R_{yc} \\ \delta p^R_{zc} \end{bmatrix}\right) \tag{15}$$

In the case of $\psi \neq 0$ Equation (15) becomes:

$$\delta\xi' = -h'\begin{bmatrix} C\psi & S\psi & 0 \\ -S\psi & C\psi & 0 \\ 0 & 0 & -\frac{r_o}{p^C_{zo}} \end{bmatrix}\left(\begin{bmatrix} \delta p^R_{xo} \\ \delta p^R_{yo} \\ \delta p^R_{zo} \end{bmatrix} - \begin{bmatrix} \delta p^R_{xc} \\ \delta p^R_{yc} \\ \delta p^R_{zc} \end{bmatrix}\right) \tag{16}$$

Where $h' = \alpha\frac{\lambda}{p^C_{zo}}$.

3. Control Problem

Recall that, our aim is moving the manipulator maintaining the image of the tracked object (its centre of gravity) coincident to the centre of the image plane, for the 2D control; and the image of the tracked object (its centre of gravity) coincident to the centre of the image plane with a given image radius, for the 3D control.
The control problem is formulated as the design of a controller which computes a control signal Δ corresponding to the movement of the robot's arm in such a way that the actual image object state reaches the desired state.

3.1 2D Control Problem Formulation

For 2D control the desired state, $\begin{bmatrix} u_d & v_d \end{bmatrix}^T$, is the centre of gravity of the object's image. The *state error* being defined as:

$$\tilde{\xi} = \xi_d - \xi = \begin{bmatrix} \tilde{u} \\ \tilde{v} \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix}$$

which could be calculated at every measurement time and used to move the robot in a direction allowing its decrease. Therefore, the control aims at ensuring that

$$\lim_{t\to\infty}\tilde{\xi} = \lim_{t\to\infty}\begin{bmatrix} \tilde{u}(t) & \tilde{v}(t) \end{bmatrix}^T = 0 \in \Re^2$$

provided that the initial feature error $\tilde{\xi}(0)$ is sufficiently small.
We make the following assumptions for the 2D control problem:

A1.0 The object is static.

A1.1 There exists a robot joint configuration $\mathbf{q}_d$ for which $\xi_d = \xi(\mathbf{q}_d)$.

A1.2 ψ is the mechanical angle between Σ_C and Σ_R, and can take different values but will be constant in each experiment.

A1.3 The axes X^R and X^C, Fig. 1, are parallels.

A1.4 The distance p^C_{xo} from the camera to the object is constant.

Assumption $A1.0$, ensures that only the control problem is evaluated. Assumption $A1.1$, ensures that the control problem is solvable. Assumption $A1.2$, stability condition, will be different for each value of ψ. Assumption $A1.3$ maintains the condition of equation (5).

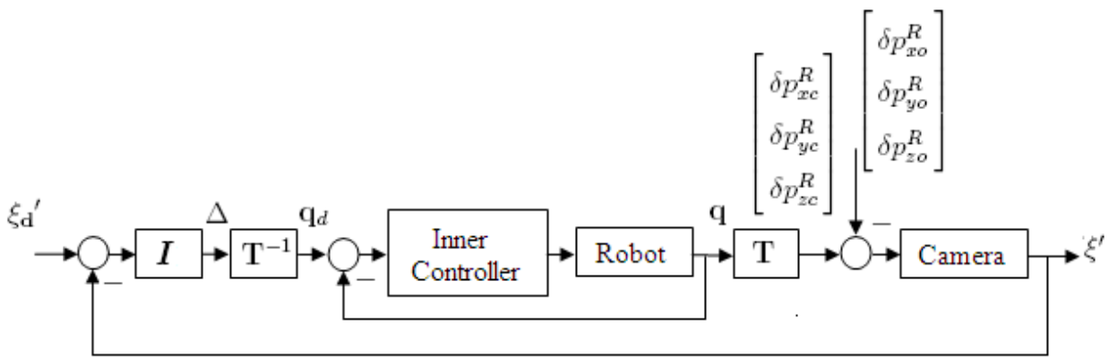


Fig. 2. Control scheme with visual feedback

3.2 3D Control Problem Formulation

In the case of 3D control the desired state object's image centre of gravity and radius is represented by $\begin{bmatrix} u_d & v_d & r_d \end{bmatrix}^T$. The *state error* is defined as

$$\tilde{\xi}' = \xi_d' - \xi' = \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{r} \end{bmatrix} = \begin{bmatrix} u_d \\ v_d \\ r_d \end{bmatrix} - \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

The control aims at ensuring that

$$\lim_{t\to\infty} \tilde{\xi}' = \lim_{t\to\infty} \begin{bmatrix} \tilde{u}(t) & \tilde{v}(t) & \tilde{r}(t) \end{bmatrix}^T = 0 \in \Re^3$$

provided that the initial feature error $\tilde{\xi}'(0)$ is sufficiently small.
We make the following assumptions the 3D control problem:

A2.0 The object is static for position regulation.

A2.1 There exists a robot joint configuration $\mathbf{q}_d$ for which $\xi_d' = \xi'(\mathbf{q}_d)$.

A2.2 The axes Z^R and Z^C, Fig. 1, are parallel.

A2.3 The initial feature error $\tilde{\xi}'(0)$ is sufficiently small.

A2.4 The object moves with low velocity in following a simple trajectory.

Assumption $A2.0$, ensures that the regulation problem is evaluated. Assumption $A2.1$, ensures that the control problem is solvable. Assumption $A2.2$, conditions for Euler angles. Assumption $A2.3$ makes possible the linear analysis about the operating point. Assumption $A2.4$ conditions of trajectory following.

3.3 Controller With Visual Feedback

For our control problem formulation, the state vector of the object can only be measured through the camera, as such, a direct knowledge of the desired joint position $\mathbf{q}_d$ is not available. Nevertheless, the desired joints position can be obtained as a result of the estimated control signal Δ and the solution of the kinematics problems.

The implemented closed-loop block diagram can be described as shown in Fig. 2. The control system has two loops in cascade, the internal loop solving the robotsŠ joint control, and the external loop implementing a dynamic *look and move* visual controller.
The inner control loop has an open control architecture; in this architecture it is possible to implement any type of controller. One possibility is to use a non-linear controller in the state variables, called torque-calculated (Kelly and Santibáñez; 2003)having the following control equation:

$$\tau = \mathbf{M}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_{vi}\dot{\tilde{q}} + \mathbf{K}_{pi}\tilde{\mathbf{q}}] + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q})$$

Where $\mathbf{K}_{pi} \in \Re^{n\times n}$ and $\mathbf{K}_{vi} \in \Re^{n\times n}$ are the symmetric positive-definite matrices and $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$. Kelly (Kelly and Santibáñez; 2003) demonstrated that with this configuration the system behaves in a closed loop as a linear multivariable system, decoupled for each robot's joint, suggesting that the matrices could be specified as:

$$\mathbf{K}_{pi} = diag\{\omega_1^2, \ldots, \omega_n^2\}$$
$$\mathbf{K}_{vi} = diag\{\omega_1, \ldots, \omega_n\}$$

In this way each joint behaves as a critically damping second order linear system with bandwidth ω_i. The bandwidth ω_i determines the speed of response of each joint. In such way the dynamic effect of the internal loop could be independent with regard to the external loop, being, according to (Hernández et al.; 2008), under the conditions that:

$$\mathbf{q}(t) = \mathbf{q}_d(t) \quad \forall t > 0 \tag{17}$$

Nevertheless, the vision-based control systems are fully sampled data systems. The feedback sensor has some dynamic characteristics such as: transport delay of pixel camera, image processing algorithms, communication between the vision system and control computer, etc. Åström (Astrom and Wittenmark; 1990) established that the sampling rate of digital control systems should be between 10 and 30 time the desired closed loop bandwidth. For the case of a 20Hz vision system the close loop bandwidth should be between 0.66 to 2 Hz.
With these conditions for the internal and external loop it is possible to make a design in order to avoid the dynamic effect of the internal loop in relation with the dynamics of the external loop (Lange and Hirzinger; 2003), a complete analysis of this topic can be found in (Hernández et al.; 2008).
But, if we analyze the control problem in the field of digital control systems, other dynamic representation of the set robot vision system could be as one or two delay units for the vision set (Corke; 1996) or for the robot (Bonfe et al.; 2002). Using this consideration we modify Equation (17) as,

$$\mathbf{q}(k) = \mathbf{q}_d(k-1) \quad \forall k > 0 \tag{18}$$

In this chapter, we consider a simpler approach, which consists of directly using the image feature vectors $\tilde{\xi}$ for 2D control or $\tilde{\xi}'$ for 3D control, being the difference between the centre of the image plane ξ_d or ξ'_d and the centre of gravity of the object in the image space ξ or ξ'; and for 3D control the difference between the desired radius in the image plane r_d and the actual radius of the object in the image plane *r (image coordinate frame)*. This error depends on the object's absolute position, in the task space $\mathbf{p}_o^R$, and the camera's centre position, $\mathbf{p}_{Corg}^R$, according to Equations (6) 2D or (16) 3D.

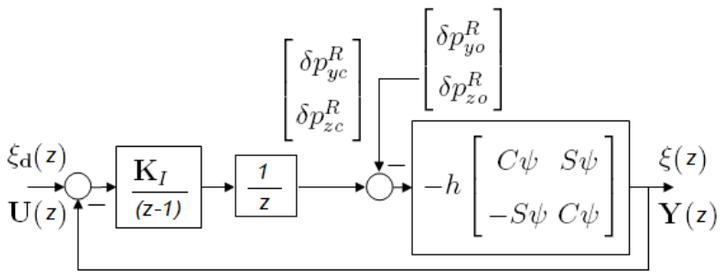


Fig. 3. 2D Vision-based simplified control scheme

4. 2D Vision-based Control. Stability Analysis

A very simple ***I*** controller can be used in this control scheme (Hernández et al.; 2008), for that case the control law can be given by:

$$\mathbf{\Delta} = \mathbf{K}_I \int \tilde{\xi} \tag{19}$$

Where $\mathbf{K}_I \in \Re^{2\times 2}$ is the symmetric integral matrix:

$$\mathbf{K}_I = \begin{bmatrix} -K_{I_1} & 0 \\ 0 & -K_{I_2} \end{bmatrix} \tag{20}$$

As in Sim's work (Sim et al.; 2002), Δ can be interpreted as the coordinates increment in the world space as a result of the image feature error $\tilde{\xi}$. Solving the inverse kinematics problem T^{-1} it is possible to obtain $\mathbf{q}_d$. The proposed system works as a regulator system, because ξ_d is constant and can be set $= 0$.

Taking into account Fig. 2, obtaining the discrete equivalence of the controller of Equation (19) and according to Equations (6) and (18); a simplified diagram can be obtained as shown in Fig. 3.

Making

$\mathbf{K} = -\alpha \frac{\lambda}{p^C_{zo}} \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} = -h \begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix}$ and $\mathbf{K}_I = \begin{bmatrix} -K_{I_1} & 0 \\ 0 & -K_{I_2} \end{bmatrix}$

according to Fig. 3 and if we consider the disturbance $\begin{bmatrix} \delta p^R_{xc} & \delta p^R_{yc} \end{bmatrix}^T$ as the system's initial conditions, the closed loop transfer function, taking a sampling period of $50ms$, can be written as:

$$\frac{0.05\mathbf{K}_I\mathbf{K}}{(z^2 - z)}[\mathbf{U}(z) - \mathbf{Y}(z)] = \mathbf{Y}(z) \tag{21}$$

where

$$\mathbf{Y}(z) = \xi(z)$$

Solving and taking the inverse Z transform we obtain:

$$\mathbf{y}(k+2) - \mathbf{y}(k+1) = -0.05\mathbf{K}_I\mathbf{K}\mathbf{y}(k) + 0.05\mathbf{K}_I\mathbf{K}\mathbf{u}(k) \tag{22}$$

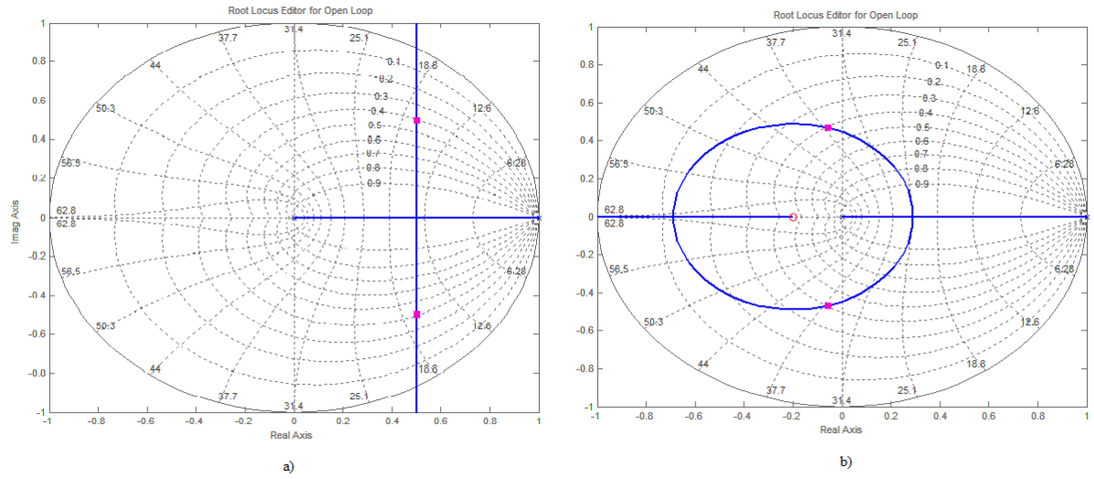


Fig. 4. Root-locus of axis model for: a) I controller with closed poles for gain $hK_{I_i} = 10$. b) PI controller with Closed poles for gain $hK_{I_i} = 30$

Taking into account that the aim is moving the manipulator maintaining the image of the tracked object (its centre of gravity) coincident to the centre of the image plane, we can make u=0 and v=0, then $\mathbf{u}$=0 and (22) becomes

$$\mathbf{y}(k+2) = \mathbf{y}(k+1) - 0.15\mathbf{K}_I\mathbf{K}\mathbf{y}(k) \tag{23}$$

The Equations (23) can be represented in the state space as

$$\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.15\mathbf{K}_I\mathbf{K} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix} \tag{24}$$

where

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.15\mathbf{K}_I\mathbf{K} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.05hC\psi K_{I_1} & -0.05hS\psi K_{I_2} & 1 & 0 \\ 0.05hS\psi K_{I_1} & -0.05hC\psi K_{I_2} & 0 & 1 \end{bmatrix}$$

In this work a robot manipulator ASEA IRB6 with camera in hand is used as case study. This type of robot has the wrist mechanically decoupled from the arm's movements, this allows to maintain the orientation of the camera. It is possible to establish ψ constant and for simplicity we set $\psi = 0$.

With this consideration and according to Equation (4) Equation (24) can be transformed into two decoupled control systems, one in each axis. The representation of these systems in the space state is:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05hK_{I_1} & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05hK_{I_2} & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \tag{26}$$

The system's stability is determined by the roots of the characteristic polynomial of matrix $\mathbf{\Phi}$. In these conditions it is easy to design the regulator for each axis.
A better illustration is shown by the root-locus diagram of the systems (25) or (26), given in Fig. 4 a), where instability is clearly indicated as the loop gain increases. If we use a PI controller, including a zero in $z = -0.2$, the diagram of Fig. 4 a) is modified as is shown in Fig. 4 b). In this case the stability condition of the system has been increased.
With the PI controller Equations (25) and (26) are modifed as:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.01hK_{I_1} & 1-0.05hK_{I_1} \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \tag{27}$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.01hK_{I_2} & 1-0.05hK_{I_2} \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \tag{28}$$

5. 3D Vision-based Control. Stability Analysis

Similar to the 2D vision-based control analysis, we use a very simple ***I*** controller in the control scheme, for that case the control law is given by:

$$\mathbf{\Delta}' = \mathbf{K}'_I \int \tilde{\xi}' \tag{29}$$

Where $\mathbf{K}'_I \in \Re^{3\times 3}$ is the symmetrical integral matrix:

$$\mathbf{K}'_I = \begin{bmatrix} -K'_{I_1} & 0 & 0 \\ 0 & -K'_{I_2} & 0 \\ 0 & 0 & K'_{I_3} \end{bmatrix} \tag{30}$$

In this case too, Δ' can be interpreted as the coordinates increment in the world space as a result of the image feature error $\tilde{\xi}'$. Solving the inverse kinematics problem T^{-1} it is possible to obtain $\mathbf{q}_d$. An analysis of this control scheme in continuous time can be found in (Hernández et al.; 2008a), where also the control performance in following simple trajectories in 3D is presented.
The system works as regulator, because ξ'_d is constant and can be set $= 0$. In these conditions, following the analysis of Section 4, using Equations (15) and (29) the system's output $\begin{bmatrix} u(k) & v(k) & r(k) \end{bmatrix}^T$, can be obtained as decoupled equations by axis, as:

$$\begin{bmatrix} v(k+1) \\ v(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05h'K'_{I_1} & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ v(k+1) \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} u(k+1) \\ u(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05h'K'_{I_2} & 1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} \tag{32}$$

$$\begin{bmatrix} r(k+1) \\ r(k+2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.05\frac{h'r_o}{p^C_{zo}}K'_{I_3} & 1 \end{bmatrix} \begin{bmatrix} r(k) \\ r(k+1) \end{bmatrix} \tag{33}$$

We considered the disturbance $\begin{bmatrix} \delta p^R_{xo} & \delta p^R_{yo} & \delta p^R_{zo} \end{bmatrix}^T$ as system initial conditions.
With $\psi \neq 0$ as operating point, for the three previous Equations, according Equation (16), only (33) is enabled. Its stability analysis can be done in a Root-locus similar to Fig. 4. For the analysis along the axes v and u Equation (16) can be simplified as:

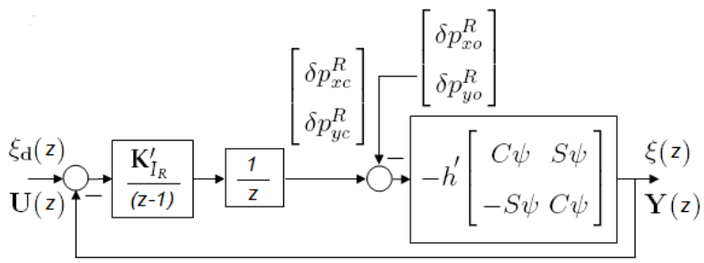


Fig. 5. 3D Vision-based simplified control scheme

$$\delta\tilde{\xi}'' = -\alpha\frac{\lambda}{p^C_{zo}}\begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix}\left(\begin{bmatrix} \delta p^R_{xo} \\ \delta p^R_{yo} \end{bmatrix} - \begin{bmatrix} \delta p^R_{xc} \\ \delta p^R_{yc} \end{bmatrix}\right) \tag{34}$$

For simplicity the δ has been avoided in Equations.
Taking into account Fig. 2 and according to Equation (17) a simplified diagram can be obtained as shown in Fig. 5.
Making

$$\mathbf{K}' = -\alpha\frac{\lambda}{p^C_{zo}}\begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix} = -h'\begin{bmatrix} C\psi & S\psi \\ -S\psi & C\psi \end{bmatrix}$$

and

$$\mathbf{K}'_{I_R} = \begin{bmatrix} -K'_{I_1} & 0 \\ 0 & -K'_{I_2} \end{bmatrix}$$

according to Fig. 5 and if we consider the disturbance $\begin{bmatrix} \delta p^R_{xo} & \delta p^R_{yo} \end{bmatrix}^T$ as the system's initial conditions, the closed loop transfer function, taking a sampling period of $50ms$, can be written as:

$$\frac{0.05\mathbf{K}'_{I_R}\mathbf{K}'}{(z^2-z)}[\mathbf{U}(z)-\mathbf{Y}(z)] = \mathbf{Y}(z) \tag{35}$$

where

$$\mathbf{Y}(z) = \tilde{\xi}''(z) \tag{36}$$

solving and taking the inverse Z transform we obtain:

$$\mathbf{y}(k+2) - \mathbf{y}(k+1) = -0.15\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{y}(k) + 0.15\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{u}(k) \tag{37}$$

In the same way as in Section 4, we can make u=0 and v=0, then $\mathbf{u}$=0 and (37) becomes

$$\mathbf{y}(k+2) = \mathbf{y}(k+1) - 0.15\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{y}(k) \tag{38}$$

The Equations (38) can be represented in the state space as

$$\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.05\mathbf{K}'_{I_R}\mathbf{K}' & \mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix}$$

and

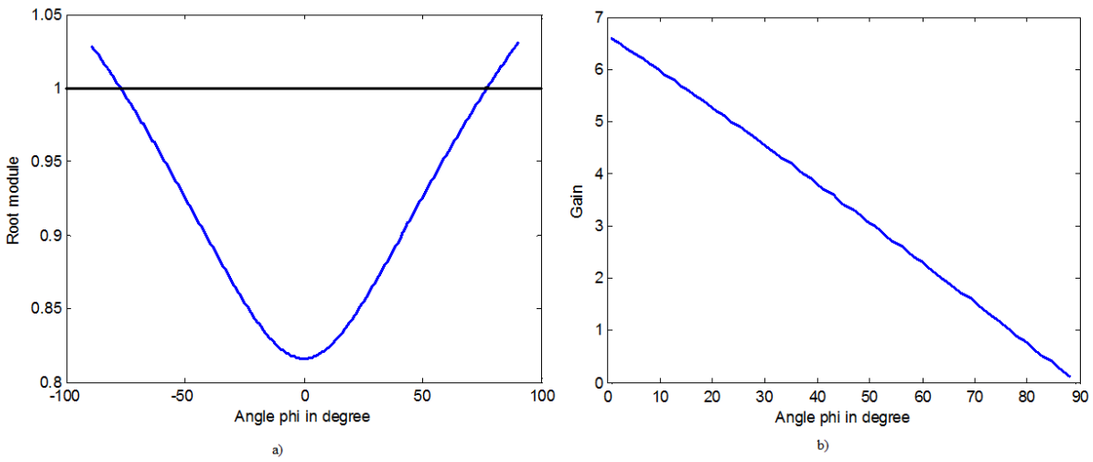


Fig. 6. a) Module of the two complex conjugated roots of the characteristic equation of I controller for $K'_{I_{1,2}} = 1$, in relation with ψ. b) Module of the angle limit for the stability of I controller in dependence of $K_{I_{1,2}}$

$$\mathbf{\Phi} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.05hC\psi K'_{I_1} & -0.05hS\psi K'_{I_2} & 1 & 0 \\ 0.05hS\psi K'_{I_1} & -0.05hC\psi K'_{I_2} & 0 & 1 \end{bmatrix}$$

The system's stability is determined by the roots of the characteristic polynomial of matrix $\mathbf{\Phi}$. If the module of any root is bigger than 1, out of unit circle, the system is not stable. An analytical solution for the roots of characteristic equation of matrix $\mathbf{\Phi}$ is very difficult in this case, but a graphic solution to illustrate the tendency of stability in relation with $\mathbf{K}'_{I_R}$ and ψ can be found. With $K'_{I_1} = K'_{I_2} = K'_{I_{1,2}} = 1$, in Fig. 6 a) the module of the two complex conjugated roots of the characteristic equation is plotted, in relation to ψ. It is clear that for $-77.4^o < \psi < 77.4^o$ the system is stable, these angle limits define the control system work space, while in Fig. 6 b) the module of the angle limit for the stability for differences values of $K'_{I_{1,2}}$ is shown.

If we use a PI controller, including a zero in $z = -0.2$ in each axis, Equation (35) is modified as:

$$\frac{0.05\mathbf{K}'_{I_R}\mathbf{K}'(z+0.2)}{(z^2-z)}[\mathbf{U}(z) - \mathbf{Y}(z)] = \mathbf{Y}(z) \tag{39}$$

and Equation (22) becomes:

$$\begin{aligned} \mathbf{y}(k+2) - (\mathbf{I} - 0.05\mathbf{K}'_{I_R}\mathbf{K}')\mathbf{y}(k+1) = \\ -0.01\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{y}(k) + 0.05\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{u}(k+1) + 0.03\mathbf{K}'_{I_R}\mathbf{K}'\mathbf{u}(k) \end{aligned} \tag{40}$$

The representation of Equation (40) in the state space,with u=0, is:

$$\begin{bmatrix} \mathbf{y}(k+1) \\ \mathbf{y}(k+2) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -0.03\mathbf{K}'_{I_R}\mathbf{K}' & \mathbf{I} - 0.15\mathbf{K}'_{I_R}\mathbf{K}' \end{bmatrix} \begin{bmatrix} \mathbf{y}(k) \\ \mathbf{y}(k+1) \end{bmatrix}$$

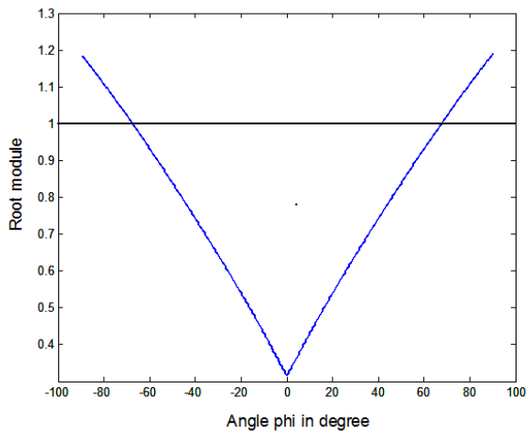


Fig. 7. Module of the two complex conjugated roots of the characteristic equation of PI controller for $K'_{I_{1,2}} = 10$, in relation with ψ

and

$$\mathbf{\Phi} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.01h'C\psi K'_{I_1} & -0.01h'S\psi K'_{I_2} & 1-0.05h'C\psi K'_{I_1} & 0.05h'S\psi K'_{I_2} \\ 0.01h'S\psi K'_{I_1} & -0.01h'C\psi K'_{I_2} & -0.05h'S\psi K'_{I_1} & 1-0.05h'C\psi K'_{I_2} \end{bmatrix}$$

With the inclusion of PI controller in each axis the stability system condition of the system is increased, as is shown in Fig. 7, where $K'_{I_{1,2}} = 10$, 10 times the gain of Fig. 6 the stability performance is similar.

6. Experimental Study

A robot manipulator ASEA IRB6 with open computer control architecture designed and built at the Departamento de Automática de la Universidad Central de Las Villas, Santa Clara, Cuba, is used as a study case. The control scheme described in Section 4, 2D vision-based control, has been implemented in the links two and three of the robot and the control scheme described in Section 5, 3D vision-based control, has been implemented in the links one, two and three of the robot. The inner loops are implemented in a PC Intel Pentium III 500 MHz connected to the robot through a Humusoft MF624 board which reads the encoder's joint position, executes the control algorithm and gives the control signal to the power unit with a sampling period of $1ms$. The video signal is acquired via a frame grabber EZ-Capture with chipset BT878 mounted on a second Intel Celeron at 2.0 GHz computer which processes the images, extracts the object's centre of gravity and radius and solves the inverse kinematic problem. Data are sent back to the main host computer during robot operation through a RS232 serial communication link, the sampling period of the external loop is $50ms$.

6.1 Practical Design Consideration

As shown in Fig. 2 the control system has two loops. The external loop calculates the image feature error at every measurement time. This control system allows the possibility of tracking

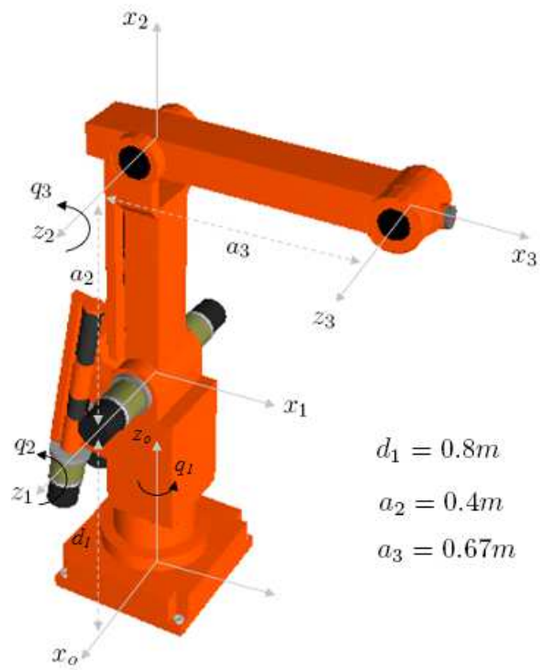


Fig. 8. Geometric description of the robotic system

Link	a_i	α_i	d_i	q_i
1	0	$\pi/2$	d_1	q_1
2	a_2	0	0	q_2
3	a_3	0	0	q_3

Table 1. Denavit-Hartenberg parameters

objects in the environment and hence using more complex vision tasks as e.g. in (Enescu et al.; 2006), for this reason a sampling period of $50ms$ has been taken for the external loop.
Åström (Astrom and Wittenmark; 1990) established that for a correct representation of time response of a continuous system, a sreasonable sampling rate is 4 to 10 samples during the rise time. With this consideration $1.5s$ of settling time is adequate for our external loop.
The internal loop does not need the same sampling period, in this case we implemented a ***P-PI*** (Sciavicco and Sciciliano; 1996) decoupled controller with $\varphi = 0.9$ and $\omega_n = 40$ as design characteristics and $1ms$ as sampling period for each robot joint.
With these conditions for the internal and external loop it is possible to avoid the dynamic effect of the internal loop in relation with the dynamic of the external loop (Lange and Hirzinger; 2003), (Hernández et al.; 2008), or make the approximation that the dynamic or the internal loop is equivalent to a time delay of the external loop, (Corke; 1996).

6.2 Simulation

Following the scheme of Fig. 2 a simulation process has been developed using MATLAB/Simulink. The Denavit-Hartenberg parameters of the robot's geometric configuration (Fig. 8) are specified in Table 1.
According to Fig. 8 the forward and inverse kinematic (1) and (2) are obtained.

For the forward kinematic:

$$p_{xc}^R = C_1(a_2C_2 + a_3C_{23})$$
$$p_{yc}^R = S_1(a_2C_2 + a_3C_{23})$$
$$p_{zc}^R = a_2S_2 + a_3S_{23} + d_1$$

Where S_{23} is $sin(q_2 + q_3)$ and, C_{23} is $cos(q_2 + q_3)$.
And for the inverse kinematic:

$$q_1 = \arctan\left(\frac{p_{yc}^R}{p_{xc}^R}\right)$$

$$q_2 = \arctan\left(\frac{p_{zc}^R - d_1}{\pm\sqrt{{p_{xc}^R}^2 + {p_{yc}^R}^2}}\right) + \arctan\left(\frac{a_3\sin(q_3)}{a_2 + a_3\cos(q3)}\right)$$

$$q_3 = \arctan\left(\frac{\pm\sqrt{1-\cos^2(q_3)}}{\cos(q_3)}\right)$$

where

$$\cos(q_3) = \frac{{p_{xc}^R}^2 + {p_{yc}^R}^2 + \left(p_{zc}^R - d_1\right)^2 - {a_2}^2 - {a_3}^2}{2a_2a_3}$$

and: $d_1 = 0.8m$, $a_2 = 0.4m$ and $a_3 = 0.67m$
For our simulation the transfer function of the DC motors of each joint is:

$$G(s)_{motor} = \frac{K_m}{s(T_ms+1)} = \frac{1550}{s(0.024s+1)}$$

In the simulation, the robot dynamic model is avoided, taking into account the slow robot velocity and that the gear reducer is $\frac{1}{150}$.
The simulation of the 3D vision-based control is presented, in this case we used the vision system model as given by Equation (16) with a sampling period of $50ms$.
We make the simulation over small variation about the operating point: $u_d = 0$, $v_d = 0$ and $r_d = 40pixels$; corresponding to object position in reference to the camera: $p_{xo}^C = 0$, $p_{yo}^C = 0$ and $p_{zo}^C = 1.10m$.
In the simulation, after guaranteeing the correct position of the camera, in the second 2, a step displacement of 0.2m in the object position is produced in all the axes, as the following mathematical representation shows.

$$p_{xo}^R = \begin{cases} 0.67m, & if \quad t < 2sec \\ 0.87m, & if \quad t \geq 2sec \end{cases}$$

$$p_{yo}^R = \begin{cases} 0.00, & if \quad t < 2sec \\ 0.20m, & if \quad t \geq 2sec \end{cases}$$

$$p_{zo}^R = \begin{cases} 2.30m, & if \quad t < 2sec \\ 2.50m, & if \quad t \geq 2sec \end{cases}$$

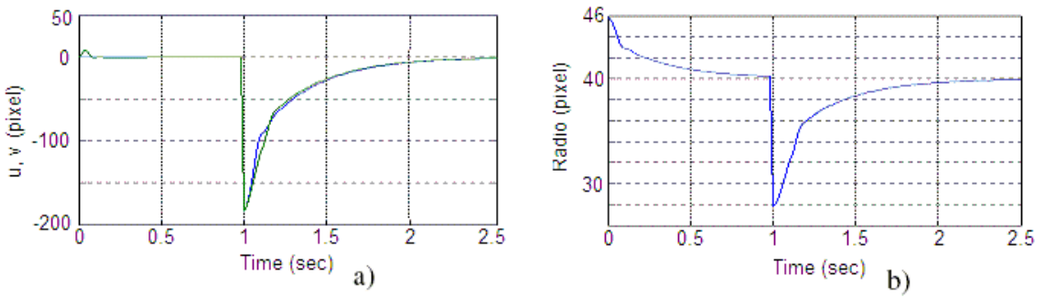


Fig. 9. Simulation of the step disturbance response in image coordinates: a) u (green) and v (blue); and b) radius

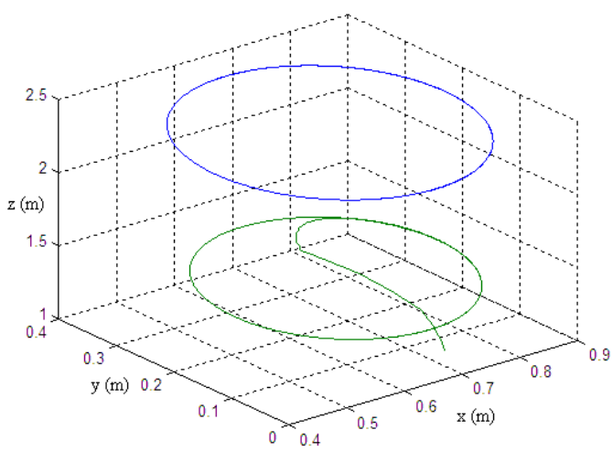


Fig. 10. Simulation of control to follow a desired trajectory

The result of the simulation for the image coordinates $\xi = \begin{bmatrix} u & v \end{bmatrix}^T$ is shown in Fig. 9 a) where it is clear that the condition $\tilde{\xi} = \begin{bmatrix} \tilde{u} & \tilde{v} \end{bmatrix}^T = \begin{bmatrix} u_d & v_d \end{bmatrix}^T - \begin{bmatrix} u & v \end{bmatrix}^T = 0$ is achieved near $1.5s$ after the disturbance and in Fig. 9 b)the same process for the image radius is shown.
As an additional test we simulated the movement of the object in a desired trajectory as an inclined ellipse in the space, as is shown in blue in Fig. 10. In green in the figure is shown the movement of the center of camera coordinate frame. The following is made with a permanent error.

6.3 Experimental Results

The control scheme proposed was implemented on the platform developed for a robot ASEA IRB6. The control algorithm of the inner loop has been implemented using MATLAB/Simulink with the *Real Time Workshop Toolbox* and *Real Time Windows Target*. For the vision loop a monochromatic camera JAI CV-252 has been used. A software component has been developed in Borland Delphi to capture the visual information and the Matrox Image Library is used to process the acquired images in real time. For our experiments, focal length $\lambda = 8mm$ and scaling factor $\alpha = 129\frac{pixels}{m}$.

Fig. 11. 2D Vision-based control, experimental set-up

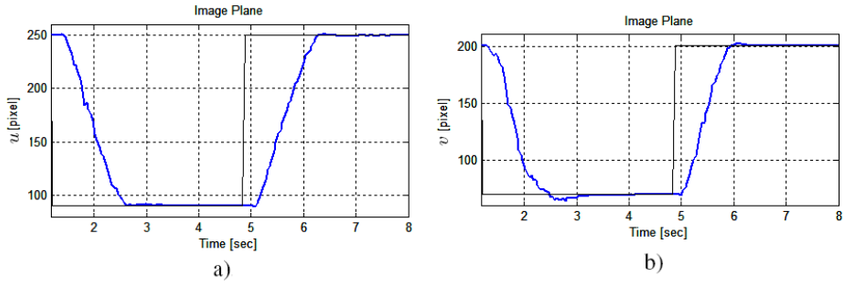


Fig. 12. Time response of image centre of gravity position in axis u a) and axis v b).

6.3.1 2D Vision-based Control.

Fig. 11 shows a view of the experimental set-up for the 2D vision-based control.
For the experiment the wanted image values changes in step for u_d between $250 pixels$ to $90 pixels$; and for v_d between $200 pixels$ to $90 pixels$.
The time responses to the steps are showed in Fig. 12 for the parameters: Fig. 12 a) shows the time response of centre of gravity image u and Fig. 12 b) shows time response of centre of gravity image v. In all the cases the final values are obtained with a good settling time and without steady state error.
Also the control to follow a desired trajectory such as a circle has been implemented. In Fig. 13 the desired trajectory and the actual trajectory are shown, in the image plane. The trajectory control has a permanent error.

6.3.2 3D Vision-based Control.

For the 3D vision-based control, Fig. 14 shows a view of the experimental set-up. For easier physical implementation, the camera axis Z^C is parallel to Z^R axis, according to assumption A2.2, but with different direction. For that reason for the rotation matrix (10), $\phi = 0$ and $\theta = \pi$ and Equation (12) becomes:

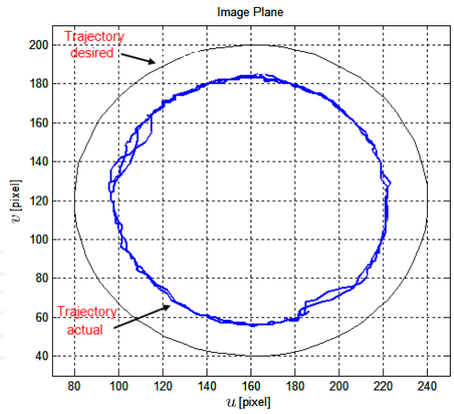


Fig. 13. Trajectory control in the image plane

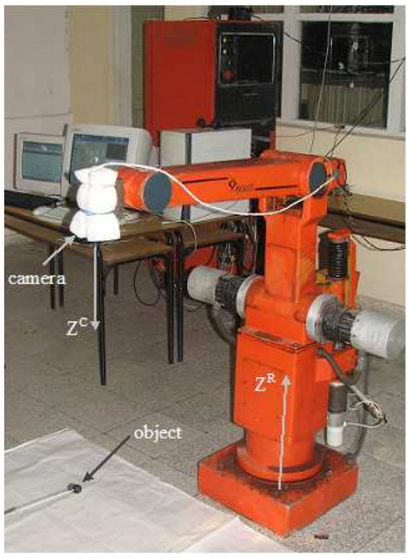


Fig. 14. 3D Vision-based control, experimental set-up

$$\mathbf{R}'^{R}_{C}(\psi) = \begin{bmatrix} -C\psi & S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{41}$$

The only consequence of this modification is the change in the theoretical sign in the regulator of axis u and z.

For the experiment the wanted image values are $u_d = 120 pixels$, $v_d = 100 pixels$ and $r_d = 30 pixels$. In $t = 0$ the actual object image features, initial conditions, are $u_0 = 30 pixels$, $v_0 = 240 pixels$ and $r_0 = 13 pixels$. The control system moves the robot camera (Tool Centre Point) as is presented in Fig. 15.

The time responses to the step in the initial conditions are shown in Fig. 16 for the three parameters: Fig. 16 a) and b) show respectively time response of centre of gravity image, u and v; and c) show the time response of image radius. In all the cases the final values are obtained with a good settling time (around 1.5sec) and without steady state error.

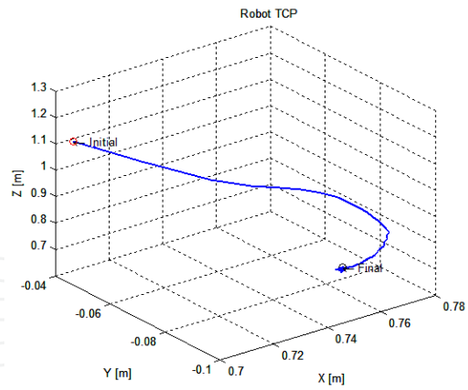


Fig. 15. Movement of the robot Tool Centre Point from the initial position to the wanted space position

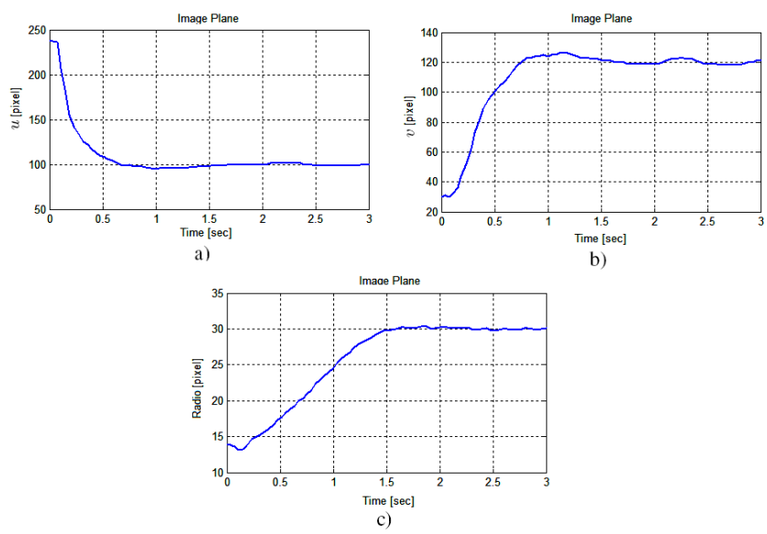


Fig. 16. Time response of image centre of gravity position in axis u a) and axis v b); and time response of image radius r c).

7. Conclusion

In this chapter we presented an image-based visual controller 2D and 3D control of camera-in-hand Robot Manipulators.

The controllers are structured in two loops, the internal loop solves the robot's joint control and in the external loop is implemented a visual controller. The dynamic effect of the internal loop is approximated as an external loop time delay. The robotic and vision systems are modeled for small variation around the operating point for position control. In these conditions the stability of the whole system in discrete time is balanced for I and PI controllers in the external loop in both cases. A particular study is made using an ASEA IRB6 robot manipulator which has its wrist mechanically decoupled. This allows maintaining the orientation of camera axis while the arm is moving. The experimental results presented illustrate the control system's stability and performance.

8. References

Astrom, K. J. and Wittenmark, B. (1990). *Compuer controlled systems: theory and design*, 2nd edn, Englewood Cliffs (NJ).

Barrientos, A., Peñín, L. F., Balguer, C. and Aracil, R. (1997). *Fundamentos de Robótica*, McGraw Hill.

Bonfe, M., Minardi, E. and Fantuzzi, C. (2002). Variable structure pid based visual servoing for robotic tracking and manipulation, *in* IEEE (ed.), *International Conference on Inteligent Robots and Systems*, Lausanne, Switzerland.

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. part i basic approaches, *IEEE Robotics and Automation Magazine* .

Chaumette, F. and Hutchinson, S. (2007). Visual servo control. part ii advanced approaches, *IEEE Robotics and Automation Magazine* .

Corke, P. I. (1996). *Visual control of robots : high-performance visual servoing*, Robotics and mechatronics series ; 2, Research Studies Press ; Wiley, Taunton, Somerset, England New York.

Corke, P. I. (2005). The machine vision toolbox. a matlab toolbox for vision and vision-based control, *IEEE Robotics and Automation Magazine* .

Enescu, V., De Cubber, G., Cauwerts, K., Sahli, H., Demeester, E., Vanhooydonck, D. and Nuttin, M. (2006). Active stereo vision-based mobile robot navigation for person tracking, *Integrated Computer-Aided Engineering* **13**: 203–222.

Flandin, G., Chaumette, F. and Marchand, E. (2000). Eye-in-hand / eye-to-hand cooperation for visual servoing, *IEEE International Conference on Robotics and Automation, ICRA2000*, San Francisco.

Hernández, L., González, R., Sahli, H., González, J. and Guerra, Y. (2008a). Simple solution for visual servoing of camera-in-hand robots in the 3d cartesian space, *10th Intl. Conf. on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, pp. 2020–2025.

Hernández, L., González, R., Sahli, H., Rubio, E. and Guerra, Y. (2008). A decoupled control for visual servoing of camera-in-hand robot with 2d movement, *in* IEEE (ed.), *Electronics, Robotics and Automotive Mechanics Conference 2008 (CERMA 2008)*, Cuernavaca, Morelos, Mexico.

Hutchinson, S., Hager, G. D. and Corke, P. I. (1996). A tutorial on visual servo control, *IEEE Transaction on Robotics and Automation* **12**(5): 651–670.

Kelly, R., Bugarin, E., Cervantes, I. and Alvarez-Ramirez, J. (2006). Monocular direct visual servoing for regulation of manipulators moving in the 3d cartesian space, *in* IEEE (ed.), *Decision and Control*, pp. 1782–1787.

Kelly, R., Carelli, R., Nasisi, O., Kuchen, B. and Reyes, F. (2000). Stable visual servoing of camera-in-hand robotic systems, *IEEE/ASME Transactions on Mechatronics* **5**(1).

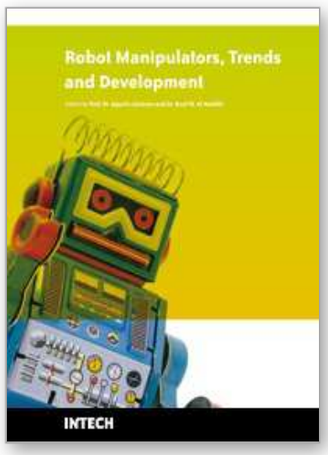
Kelly, R. and Santibáñez, V. (2003). *Control de Movimiento de Robots Manipuladores*, Pearson Education, Madrid.

Lange, F. and Hirzinger, G. (2003). Predictive visual tracking of lines by industrial robots, *The International Journal of Robotics Research* **22**.

Sciavicco, L. and Sciciliano, B. (1996). *Modeling and Control of Robot Manipulators*, McGraw Hill Co, New York.

Sim, T. P., S., H. G. and Lim, K. B. (2002). A pragmatic 3d visual servoing system, *International Conference on Robotics and Automation*, IEEE, Washington, DC.

Xie, H., Sun, L., Rong, W. and Yuan, X. (2005). Visual servoing with modified smith predictor for micromanipulation tasks, *in* IEEE (ed.), *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Canada.



Robot Manipulators Trends and Development

Edited by Agustin Jimenez and Basil M Al Hadithi

ISBN 978-953-307-073-5
Hard cover, 666 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book presents the most recent research advances in robot manipulators. It offers a complete survey to the kinematic and dynamic modelling, simulation, computer vision, software engineering, optimization and design of control algorithms applied for robotic systems. It is devoted for a large scale of applications, such as manufacturing, manipulation, medicine and automation. Several control methods are included such as optimal, adaptive, robust, force, fuzzy and neural network control strategies. The trajectory planning is discussed in details for point-to-point and path motions control. The results in obtained in this book are expected to be of great interest for researchers, engineers, scientists and students, in engineering studies and industrial sectors related to robot modelling, design, control, and application. The book also details theoretical, mathematical and practical requirements for mathematicians and control engineers. It surveys recent techniques in modelling, computer simulation and implementation of advanced and intelligent controllers.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen