KINEMATIC AND DYNAMIC 2D VISUAL SERVOING

Paulo Jorge Sequeira Gonçalves

Escola Superior de Tecnologia de Castelo Branco
Av Empresário, 6000-767 Castelo Branco
fax: +351 272339399
e-mail: paulo.goncalves@est.ipcb.pt

†ECC2001 Conference
e-mail: ecc2001@fe.up.pt
http://www.fe.up.pt/ecc2001/

Keywords: Visual Servoing, Machine-Vision, Robotic Manipulator, Robot Control.

Abstract

In this paper a comparison of kinematic and dynamic image-based (2D) visual servoing is accomplished. This comparison is based on a 2 dof planar robot manipulator constructed at Instituto Superior Técnico, and the simulation results are obtained using a Matlab 6.0 Simulink model under regulator control. For kinematic visual servoing, a control scheme with exponential decay is derived. For dynamic visual servoing, PD and inverse dynamics control were used. Finally, comparison results are presented.

1 Introduction

In this paper we control a two degrees of freedom planar robotic manipulator, based on 2D visual servoing. The robot in use was constructed at Instituto Superior Técnico [1,8], Mechanical Engineering Department, Robotics and Automation Laboratory.

Planar robots are wide spread in industry, and two of their most important characteristics are speed and precision. This type of robotic structure can easily evolve to a SCARA robot, used in vertical assembly tasks, palletising, etc.

The objectives of this work are to study the referred robotic manipulator under kinematic and dynamic 2D visual servoing. The robot should move from an initial to a final point in the image features space. To accomplish these objectives, the control laws for kinematic and dynamic visual servoing will be derived and the control structures defined. Finally the simulation results are presented and some conclusions of this work are drawn.

2 Theoretical Framework

Machine Vision and Robotics can be used together to control, a robot manipulator. This type of control defined as Visual Servoing, uses visual information from the work environment to control a robot manipulator performing some task.

The visual information can be obtained by two ways: using direct information from the image (2D visual servoing); or using 3D information of the object from the image(s) (3D visual servoing). The second case needs an on-line processing for pose calculation. A good explanation of the differences is done in [12]. In this paper only 2D visual servoing is used.

Image features are needed in order to move the robot manipulator when performing some task. So it is necessary to define:

s, image features vector;

s_d, desired image features vector;

e, error between s and s_d at some time t.

$$e = s - s_d \tag{1}$$

Following the approach to Visual Servoing made in [4], to obtain a "good control" it is necessary to minimise a function *f(s)*:

$$f(s) = \frac{1}{2} \cdot \left\| s - s_d \right\|^2 \tag{2}$$

and when the desired image features are obtained: $e = 0$.

Considering r, the pose of the end-effector (translation and rotation), which is dependent of the robot joint variables q, it is then necessary to find a robot pose that minimise *f(s)*:

$$\min_q f(q) \tag{3}$$

The function *f(s)*, reaches a minimum when it's first derivate goes to zero, so using equation (3):

$$\frac{\partial f}{\partial s} \cdot \frac{\partial s}{\partial r} \cdot \frac{\partial r}{\partial q} = 0 \tag{4}$$

$\frac{\partial s}{\partial r} \equiv J_i(r,z)$, is the image jacobian.

$$\dot{s} = J_i(r,z) \cdot \dot{r} \tag{5}$$

$\frac{\partial r}{\partial q} \equiv J_R(q)$, is the robot jacobian.

$$\dot{r} = J_R(q)\cdot\dot{q} \tag{6}$$

According to [5], if the two jacobians are not singular the solution to the minimisation problem becomes:

$$\frac{\partial f}{\partial s} = 0 \text{ , i.e., } e = 0$$

and if at the desired position the object velocity is zero, i.e., $\dot{s}_d = 0$:

$$\frac{\partial e}{\partial t} = \frac{\partial s}{\partial t} = \frac{\partial s}{\partial r}\cdot\frac{\partial r}{\partial t} \tag{7}$$

where:

$\dfrac{\partial r}{\partial t} \equiv \dot{r}$, the end effector velocity

As seen in equation (7), to obtain a control law is necessary to obtain a relation between the image features and the image features velocities. By definition of forward finite differences, [11], the image features velocity becomes:

$$\frac{\partial s}{\partial t} = \frac{1}{h}\cdot\left(s_{t+h} - s_t\right) + \bigcirc(h^2) \tag{8}$$

that can be approximated, in order to obtain our goal:

$$\dot{s} = \frac{1}{h}\cdot\left(s_d - s\right) \tag{9}$$

where h, is a constant that results from the sample time. It can be seen that equation (9), produces some errors especially if the desired features, s_d, are far away from the actual position, s. Moreover, this equation is only valid locally.

From equations (7) and (9), the following relation can be obtained:

$$\frac{\partial e}{\partial t} = J_i(r,z)\cdot\dot{r} \Rightarrow s - s_d = -h\cdot J_i(r,z)\cdot\dot{r}$$

Considering that z is constant or can be obtained, we finally reach the control law:

$$\dot{r} = -(s - s_d)\cdot\frac{1}{h}\cdot J_i^{-1}(r) \tag{10}$$

The image jacobian is calculated using the Pin-Hole camera model [3], other methods can be used as in [2]. The robot jacobian is calculated using the methods in [13].

2.1 Kinematic Visual Servoing

Following the work in [6], the end-effector pose r, can be obtained directly:

$$r = -\int\frac{1}{h}\cdot e\cdot J_i^{-1}(r)\,dt$$

The exponential decayment of the error function (2) can be verified, using equations (7) and (9):

$$\dot{e} = -K\cdot e \tag{11}$$

where:

$$K = \frac{1}{h} \tag{12}$$

The condition expressed in equation (11), that other authors stated as a condition [4,6,9,10], here is stated as a result when deriving the control law. This method makes very clear the verification that this condition is only valid in a neighbourhood, function of the sample time h, of the actual image feature position, s.

Then, the visual control law is:

$$r = \int\dot{e}\cdot J_i^{-1}(r)\,dt \tag{13}$$

The inner robot controller has implemented a PID control law, with the end-effector pose r, as input.

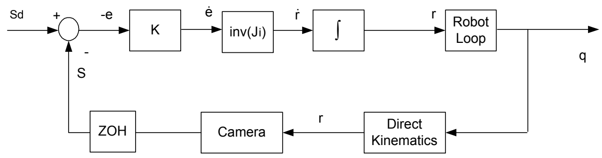

Fig. 1 Kinematic Visual Servoing Control Architecture

2.2 Dynamic Visual Servoing

For Dynamic Visual Servoing two control arquitectures were used: PD control and Inverse Kinematics control.

The first control law used was inspired in a classical PD control on the end-effector coordinates r, [13].

Considering $\dot{r}_d = 0$, using the forward finite differences as in equation (9) and $e_r = r - r_d$, the next equation arises:

$$e_r = -h\cdot\dot{r} \tag{14}$$

A final relation can be obtained from equations (1),(10) and (14):

$$e_r = J_i^{-1}\left(r\right)\cdot e \tag{15}$$

Finally, from equations (5) and (15):

$$\tau = J_R^T\left(q\right)\cdot\left[K_P\cdot J_i^{-1}\left(r\right)\cdot e - K_D\cdot J_i\left(r\right)\cdot J_R\left(q\right)\cdot\dot{q}\right] \tag{16}$$

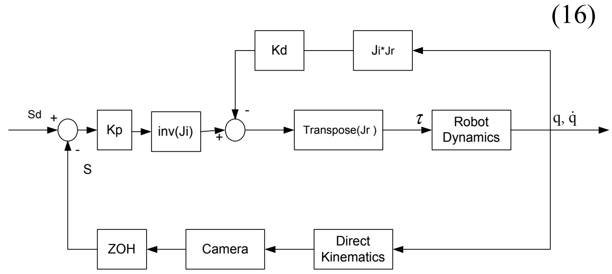

Fig. 2 PD Dynamic Visual Servoing Control Architecture

The second control law used was inspired in a classical inverse dynamics control on the end-effector coordinates *r*, [13].

Considering $\dot{r}_d = 0$, and using equations (14) and (15) the visual control law becomes:

$$\begin{aligned} \tau &= B(q)\cdot y + n(q,\dot{q}) \\ y &= J_R^{-1}(q)\cdot\left[K_P\cdot J_i^{-1}(r)\cdot e - J_i(r)\cdot\dot{J}_R(q)\cdot\dot{q}\right] \end{aligned} \quad (17)$$

where *B(q)* is the robot inertia matrix and $n(q,\dot{q})$ the coriolis and centrifugal vector.

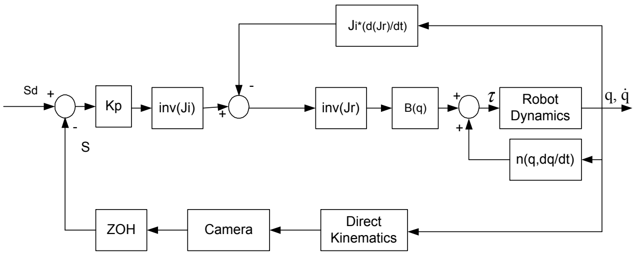

Fig. 3 Inverse Dynamics control, Dynamic Visual Servoing Control Architecture

3 Simulation results

A planar robotic manipulator of two degrees of freedom, constructed at the Mechanical Engineering Department in Instituto Superior Técnico, [1,8], was modelled in [6] to implement visual servoing algorithms.

Fig. 4 Planar Robotic Manipulator

A Matlab™ 6.0 model was developed and validated in [6]. In this section are presented the results for a feature point under regulator control, with initial position [230 -320] and final position [0 -150].

For Dynamic Visual Servoing using an Inverse Dynamics visual controller the gain with best performance was:

$$K_P = \begin{bmatrix} 2000 & 0 \\ 0 & 2000 \end{bmatrix}$$

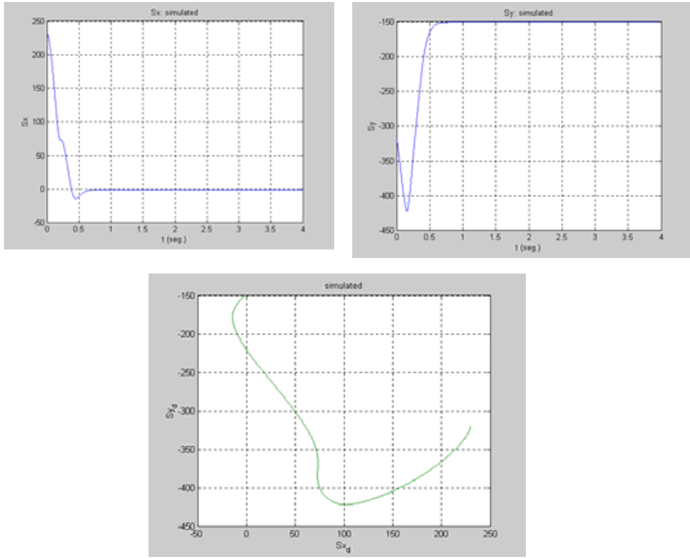

Fig. 5 Dynamic Visual Servoing – Inverse Dynamics control - Image feature response: X, coordinate; Y, coordinate; X versus Y.

For Dynamic Visual Servoing using an PD visual controller the gains with best performance were:

$$K_P = \begin{bmatrix} 1000 & 0 \\ 0 & 500 \end{bmatrix} \qquad K_D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

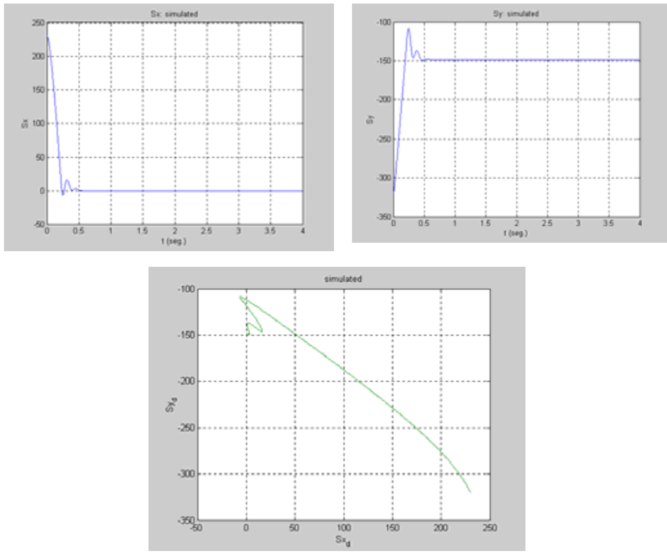

Fig. 6 Dynamic Visual Servoing – PD control - Image feature response: X, coordinate; Y, coordinate; X versus Y.

For Dynamic Visual Servoing using an PD visual controller the gain with best performance was: *K=0.*

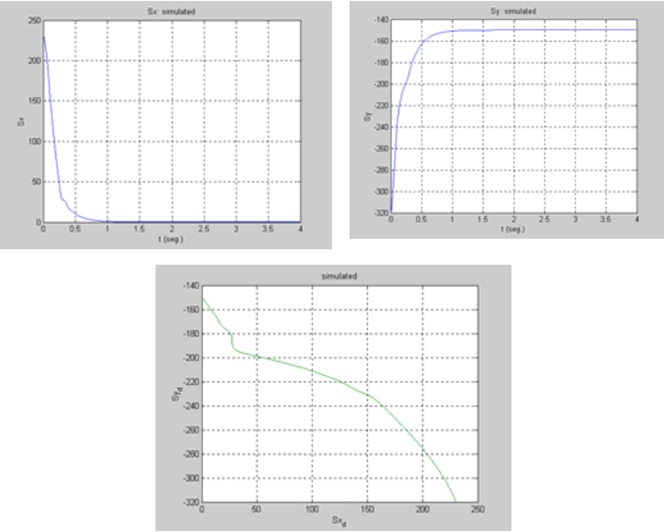

Fig. 7 Kinematic Visual Servoing Image feature response: X, coordinate; Y, coordinate; X versus Y.

4 Conclusions

In this paper 2D Visual Servoing was implemented and simulated on a validated experimental apparatus. A simple modification, when deriving the kinematic control equations, was used to clarify the locally condition and the exponential decay of the error function. For dynamic visual servoing a PD and inverse dynamics control schemes were implemented.

According to the results showed in the last section, Dynamic Visual Servoing based on a inverse dynamics controller suggests the best results for regulator control, because of the fast response time and few oscillations.

Acknowledgements

This work is supported by the Programa de Financiamneto Plurianual de Unidades de I&D (POCTI), do Quadro Comunitário de Apoio III. I would like to thank Luís Baptista from Escola Náutica Infante D. Henrique.

References

[1] Baptista L., “Controlo adaptativo de posição e força de robôs manipuladores”, PhD dissertation, Technical University of Lisbon, Instituto Superior Técnico, Lisbon, (2000).

[2] Chaumette F.. “Modelling Issues in Visual Servoing”. 5th Symp. on Pattern Recognition, SIARP 2000, pp. 3-18, Lisbon, Portugal, (2000).

[3] Corke P.. “Visual Control of Robots”, Wiley, (1996).

[4] Espiau B., F. Caumette, P. Rives. ”A New Approach to Visual Servoing in Robotics”, *IEEE Transactions on Robotics and Automation*, **vol. 8, no. 3**, pp. 313-326, (1992).

[5] Espiau B.. ”Sur les erreurs en Asservissement Visuel”, *Rapport de Recherche nº2619*, INRIA, (1995).

[6] Gonçalves P., Pinto J.. “Experiments on 2D Visual Servoing of a 2dof Planar Robot Manipulator”, *Festival Nacional de Robótica – Encontro Científico*, Guimarães, (2001).

[7] Hager G.. “Modular System for Robust Hand-Eye Coordination Using Feedback from Stereo Vision”, *IEEE Transactions on Robotics and Automation*, **vol. 13, no. 4**, pp. 582-595, (1997).

[8] Martins J. and J. Sá da Costa. “A Methodology for the identificatoin of a two degrees of freedom robotic manipulator arm”, Procedings of the 18th IASTED international conference, Modelling, Identification and Control, pp. 412-415, Innsbruck, Austria, (1999).

[9] Malis E.. “Contributions à la modelisation et à la comande en asservissement visuel“, Phd Thesis, Univérsité de Rennes I, (1998).

[10] Oh P., P. Allen. “Design of a Partitioned Visual Feedback controller”, *ICRA98*, (1998).

[11] Pina H.. “Métodos Numéricos”, McGraw-Hill, (1996).

[12] Hutchinson S., G. Hager and P. Corke. “A Tutorial on Visual Servo Control”*, IEEE Transactions on Robotics and Automation*, **vol. 12, no. 5**, 651-670, (1996).

[13] Sciavicco L. and B. Siciliano. “Modelling and Control of Robot Manipulators”, 2nd edition, Springer-Verlag, (2000).