Fuzzy Model Based Control Applied to Path Planning Visual Servoing

Paulo J. Sequeira Gonçalves[1], Luís F. Mendonça[2],
João Costa Sousa[2], and João Rogério Caldas Pinto[2]

[1] Instituto Politécnico de Castelo Branco
Escola Superior de Tecnologia
Dept. of Industrial Engineering
6000-767 Castelo Branco, Portugal
pgoncalves@est.ipcb.pt
[2] Technical University of Lisbon,
Instituto Superior Técnico
Dept. of Mechanical Engineering, GCAR/IDMEC
1049-001 Lisboa, Portugal
{mendonca,j.sousa,jcpinto}@dem.ist.utl.pt

Abstract. A new approach to eye-in-hand path planning image-based visual servoing based on fuzzy modeling and control is proposed in this paper. Fuzzy modeling is applied to obtain an inverse model of the mapping between image features velocities and joints velocities, avoiding the necessity of inverting the Jacobian. The inverse model is directly used as a controller. The control scheme is applied to a robotic manipulator performing visual servoing, for a given profile of planned image features velocities. The obtained results show the effectiveness of the proposed control scheme.

1 Introduction

In eye-in-hand image-based visual servoing, the Jacobian plays a decisive role in the convergence of the control, due to its analytical model dependency on the selected image features. Moreover, the Jacobian must be inverted on-line, at each iteration of the control scheme. Nowadays, the research community tries to find the right image features to obtain a diagonal Jacobian [10]. The obtained results only guarantee the decoupling from the position and the orientation of the velocity screw. This is still a hot research topic, as stated very recently in [10]. In this paper, the previous related problems in the Jacobian are addressed using fuzzy techniques, to obtain a controller capable to control the system. First, the desired trajectory is planned in the image space. Second, a fuzzy model to derive the inverse model of the robot is used to compute the joints and end-effector velocities in a straightforward manner. A two degrees of freedom planar robotic manipulator is controlled, based on eye-in-hand image-based visual servoing using fuzzy control systems. The paper starts by describing briefly the concept of image-based visual servoing and path planning. Fuzzy modeling and identification is described next. The obtained results and finally the conclusions and the possible future research are discussed.

A. Sanfeliu et al. (Eds.): CIARP 2004, LNCS 3287, pp. 224–231, 2004.

2 Image-Based Visual Servoing

In image-based visual servoing, the choice of different image features induces different control laws, and its number depends also on the number of degrees of freedom (DOF) of the robotic manipulator under control. The robotic manipulator used as test-bed in this paper is depicted in Fig. 1, and it has 2 DOF. Thus, the image features s consist of the coordinates x and y of one image point, which are needed to perform the control.

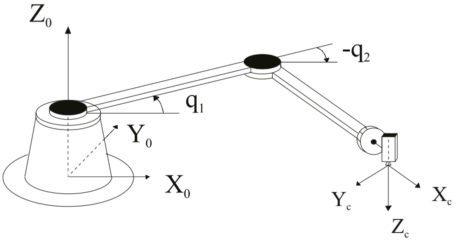


Fig. 1. Planar robotic manipulator with eye-in-hand looking down.

2.1 Modeling the Image-Based Visual Servoing System

Image-based visual servoing is used in an eye-in-hand system [6], where the camera is fixed at the robotic manipulator end-effector. The kinematic modeling of the transformation between the image features velocities and the joints velocities must be found. The kinematic modeling of the transformation between the image features velocities, $\dot{s}$, and the joints velocities $\dot{q}$ is defined as follows [4]:

$$\dot{s} = J(x, y, Z, q) \cdot \dot{q}\,, \tag{1}$$

where J is the total Jacobian, defined as:

$$J(x, y, Z, q) = J_i(x, y, Z) \cdot {}^cW_e \cdot {}^eJ_R\,(q) \tag{2}$$

where Z is the depth between the camera and object frames; $J_i(x, y, Z)$ is the image Jacobian; ${}^cW_e$ is defined as the transformation between the camera and end-effector frames velocities and ${}^eJ_R$ is the robot Jacobian for the planar robotic manipulator, [4].

2.2 Controlling the Image-Based Visual Servoing System

One of the classic control scheme of robotic manipulators using information from the vision system, is presented in [3]. The global control architecture is shown in Fig. 2, where the block *Robot inner loop* law is a PD control law, with sample time of 1 ms. The robot joint velocities $\dot{q}$ to move the robot to a predefined point in the image, s^* are derived using the *Visual control law*, [4], where an exponential decayment of the image features error is specified:

$$\dot{q} = -K_p \cdot \hat{J}^{-1}(x, y, Z, q) \cdot (s - s^*)\,. \tag{3}$$

K_p is a positive gain, that is used to increase or decrease the decayment of the error velocity.

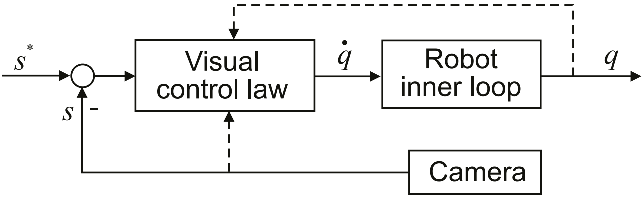


Fig. 2. Control loop of image-based visual servoing.

2.3 Visual Servoing by Path Planning

The path planning scheme presented follows the work in [7]. An image features trajectory is computed off-line, in order to be introduced on the visual control loop as the reference to be followed. The method for obtaining the image features trajectory is divided in three main parts. The first one is the data initialization, in which the initial pose is obtained from the initial and desired robot position. The second part consists of a iteration process in which the image features discrete path is composed. The third part consists of obtaining the final trajectory of the image features over time. During initialization, the image points corresponding to the image features as seen in the initial and desired robot position are obtained. The translational vector and rotational matrix between the object and the initial/desired position are computed using a pose estimation method [2]. The initialization ends with the computation of the initial pose relative to the desired one, from the resulting data. The iteration process in which the image features discrete path is composed begins with the estimation of the next pose, $\varUpsilon_{k+1}$, from the actual pose, $\varUpsilon_k$, and a composed force, F. The composed force is a sum of two different forces, an attractive, F_a, and a repulsive, F_r. For details see [7].

$$\begin{array}{l} \varUpsilon_{k+1} = \varUpsilon_k + \varepsilon_k \cdot \frac{F(\varUpsilon_k)}{\|F(\varUpsilon_k)\|}, \varepsilon_k > 0 \\ F\left(\varUpsilon_k\right) = F_a\left(\varUpsilon_k\right) + F_r\left(\varUpsilon_k\right) \end{array} \tag{4}$$

where ε_k is a positive scaling factor denoting the length of the k^{th} increment. The attractive force minimizes the trajectory and the repulsive force is set to avoid, only, the physical image limits. All the remaining poses until the robot reaches the desired position are estimated in a recursive process, which begins with the initial pose. The attractive force is highest in the initial pose and it will diminish during the trajectory, tending to zero at the final pose. The recursion ends when the attractive potential reaches a certain value close to zero, meaning that the actual pose is close enough to the desired position. The result of the iteration process is a vector of image features which contains the image variation of each image feature as the camera goes from the initial to the desired position. The final trajectory of the image features over time, s^p, must be continuous and differentiable, so was chosen a C^2 curve. It is computed from the vector obtained during the iteration process, a pre-defined sample time - video rate $\varDelta T = t_k - t_{k-1}$, and the final time for the servoing, using a cubic spline.

$$s^p\left(t\right) = A_k \cdot t^3 + B_k \cdot t^2 + C_k \cdot t + D_k \tag{5}$$

where: $(k-1) \cdot \mathit{\Delta} T \leq t \leq (k) \cdot \mathit{\Delta} T$. The control law used follows [7].

$$\dot{q} = -\lambda \cdot J^{-1} \cdot (e_p) + K_2 \cdot J^{-1} \cdot (e_{pp}) \tag{6}$$

where, $K_2 = \frac{\lambda}{\mathit{\Delta} T}$; $e_p = s_k - s_k^p$; $e_{pp} = s_k^p - s_{k-1}^p$.

2.4 Problem Statement

To derive an accurate global Jacobian, J, a perfect modeling of the camera, the image features, the position of the camera related to the end-effector, and the depth of the target related to the camera frame must be accurately determined. Even when a perfect model of the Jacobian is available, it can contain singularities, which hampers the application of a control law. To overcome these difficulties, a new type of differential relationship between the features and camera velocities was proposed in [9]. This approach estimates the variation of the image features, when an increment in the camera position ${}^cP$ is given, by using a relation G. This relation is divided into G_1 which relates the position of the camera to the image features, and F_1 which relates their respective variation:

$$s + \delta s = G({}^cP + \delta^c P) = G_1({}^cP) + F_1({}^cP, \delta^c P). \tag{7}$$

Considering only the variations in (7):

$$\delta s = F_1({}^cP, \delta^c P), \tag{8}$$

let the relation between the camera position variation $\delta^c P$, the joint position variation, δq and the previous position of the robot q be given by:

$$\delta^c P = F_2(\delta q, q). \tag{9}$$

The two previous equations can be composed because the camera is rigidly attached to the robot end-effector, i.e., knowing q, ${}^cP$ can easily be obtained from the robot direct kinematics. Thus, an inverse function F^{-1} stating that the joint velocities depends on the image features velocities and the previous position of the robot manipulator can be obtained and discretized as follows:

$$\delta q(k) = F_k^{-1}(\delta s(k+1), q(k)). \tag{10}$$

In image-based visual servoing, the goal is to obtain a joint velocity, $\delta q(k)$, capable of driving the robot according to a desired image feature position, $s(k+1)$, with an also desired image feature velocity, $\delta s(k+1)$, from any position in the joint spaces. This goal can be accomplished by modeling the inverse function F_k^{-1}, using inverse fuzzy modeling as presented in Section 3. This new approach to image-based visual servoing allows to overcome the problems stated previously regarding the Jacobian inverse, the Jacobian singularities and the depth estimation, Z.

3 Fuzzy Control Strategies

3.1 Fuzzy Modeling

Fuzzy models can be identified based entirely on systems measurements. In the following, we consider data-driven modeling based on fuzzy clustering [1]. We consider rule-based models of the Takagi-Sugeno (TS) type. It consists of fuzzy rules which each describe a local input-output relation, typically in a linear form:

$$R_i : \textbf{If } x_1 \text{ is } A_{i1} \textbf{and } \ldots \textbf{and } x_n \text{ is } A_{in} \textbf{then } y_i = \mathbf{a}_i \mathbf{x} + b_i \,, \quad i = 1, 2, \ldots, K.$$

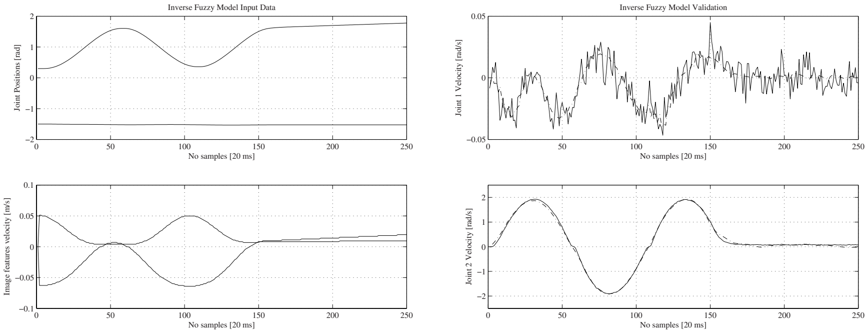
Here R_i is the ith rule, $\mathbf{x} = [x_1, \ldots, x_n]^T$ are the input (antecedent) variable, $A_{i1}, \ldots,$ A_{in} are fuzzy sets defined in the antecedent space, and y_i is the rule output variable. K denotes the number of rules in the rule base. The aggregated output of the model, $\hat{y}$, is calculated by taking the weighted average of the rule consequents: $\hat{y} = \sum_{i=1}^{K} \beta_i y_i / \sum_{i=1}^{K} \beta_i$, where β_i is the degree of activation of the ith rule: $\beta_i = \Pi_{j=1}^{n} \mu_{A_{ij}}(x_j)$, $i = 1, \ldots, K$, and $\mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0, 1]$ is the membership function of the fuzzy set A_{ij} in the antecedent of R_i. To identify the model in (11), the regression matrix X and an output vector $\mathbf{y}$ are constructed from the available data: $\mathrm{X}^T = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{y}^T = [y_1, \ldots, y_N]$, where $N \gg n$ is the number of samples used for identification. The number of rules, K, the antecedent fuzzy sets, A_{ij}, and the consequent parameters, $\mathbf{a}_i, b_i$ are determined by means of fuzzy clustering in the product space of the inputs and the outputs [1]. Given the data set and an estimated number of clusters K, the Gustafson-Kessel fuzzy clustering algorithm [5] is applied to compute the fuzzy partition matrix U. The fuzzy sets in the antecedent of the rules A_{ij} are obtained from the partition matrix U, whose ikth element $\mu_{ik} \in [0, 1]$ is the membership degree of the data object $\mathbf{z}_k$ in cluster i. The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let $\theta_i^T = \left[\mathbf{a}_i^T ; b_i\right]$, let X_e denote the matrix $[\mathrm{X}; \mathbf{1}]$ and let W_i denote a diagonal matrix in $\mathbb{R}^{N \times N}$ having the degree of activation, $\beta_i(\mathbf{x}_k)$, as its kth diagonal element. Assuming that the columns of X_e are linearly independent and $\beta_i(\mathbf{x}_k) > 0$ for $1 \leq k \leq N$, the weighted least-squares solution of $\mathbf{y} = \mathrm{X}_e \theta$ becomes $\theta_i = \left[\mathrm{X}_e^T \mathrm{W}_i \mathrm{X}_e\right]^{-1} \mathrm{X}_e^T \mathrm{W}_i \mathbf{y}$. For the robotic application in this paper, the inverse model is identified using input-output data from the inputs $\dot{q}(k)$, outputs $\delta s(k+1)$ and the state of the system $q(k)$. A commonly used procedure in robotics is to learn the trajectory that must be followed by the robot. From an initial position, defined by the joint positions, the robotic manipulator moves to the predefined end position, following an also predefined trajectory, by means of a PID joint position controller. This specialized procedure has the drawback of requiring the identification of a new model for each new trajectory. However, this procedure revealed to be quite simple and fast. Moreover, this specialized identification procedure is able to alleviate in a large scale the problems derived from the close-loop identification procedure.

4 Results

This section presents the simulation results obtained for the robotic manipulator. First, the identification of the inverse fuzzy model of the robot is described. Then, the control results using the fuzzy model based controller introduced in this paper are presented.

4.1 Inverse Fuzzy Modeling

In order to apply the controller described in this paper, first an inverse fuzzy controller must be identified. Recall that a model must be identified for each trajectory. The profile chosen for the image features velocity was derived for the planned features path, that moves the robot from the initial joints position $q = [-1.5; 0.3]$ to the final position $q = [-1.51; 1.52]$, in one second, starting and ending with zero velocity. An inverse fuzzy model (10) for this trajectory is identified using the fuzzy modeling procedure described in Section 3.1. The measurements data is obtained from a simulation of the planar robotic manipulator eye-in-hand system. The set of identification data used to build the inverse fuzzy model contains 250 samples, with a sample time of 20ms. Figure 3a presents the input data, which are the joint positions $q_1(k)$ and $q_2(k)$, and the image features velocities $\delta s_x(k)$ and $\delta s_y(k)$, used for identification. Note that to identify the inverse model, one cannot simply feed the inputs as outputs and the outputs as inputs. Since the inverse model (10) is a non-causal model, the output of the inverse model must be shifted one step, see [8]. The validation of the inverse fuzzy model is shown in Fig. 3b, where the joint velocities δq are depicted. Note that two fuzzy models are identified, one for each velocity. It is clear that the model is quite good. Considering, e.g. the performance criteria *variance accounted for* (VAF), the models have the VAFs of 70.2% and 99.7%. When a perfect match occur, this measure has the value of 100%. Then, the inverse model for the joint velocity δq_2 is very accurate, but the inverse model for δq_1 is not so good. This was expectable as the joint velocity δq_1 varies much more than δq_2. However, this model will be sufficient to obtain an accurate controller, as is shown in Section 4.2. In terms of parameters, four rules (clusters) revealed to be sufficient for each output, and thus the inverse fuzzy model has 8 rules, 4 for each output, δq_1 and δq_2. The clusters are projected into the product-space of the space variables and the fuzzy sets A_{ij} are determined.



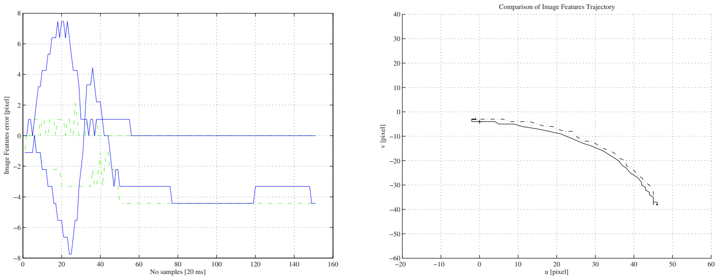
(a) Input data for fuzzy identification. Top: joint positions, q. Bottom: image feature velocity, δs.

(b) Validation of joint velocities (δq). Solid – real output data, and dash-dotted – output of the inverse fuzzy model.

Fig. 3. Input data for fuzzy modeling and validation of the obtained inverse fuzzy model.

4.2 Control Results

This section presents the obtained control results, using the classical image-based path planning visual servoing presented in Section 2, and the inverse fuzzy model-based control scheme presented in Section 3. The implementation was developed in a simulation of the planar robotic manipulator eye-in-hand system. This simulation was developed and validated in real experiments, using classic visual servoing techniques, by the authors.



(a) Image features errror, e_p.

(b) Image features trajectory s, in the image plane.

Fig. 4. Comparison between fuzzy and classical path planning visual servoing, image errors and trajectories. Solid – inverse fuzzy model control, and dash-dotted – classical visual servo control.

The comparison of the image features trajectory for both the classic and the fuzzy visual servoing controllers is presented in Fig. 4a. In this figure, it is shown that the classical controller follows the trajectory with a better accuracy than the fuzzy controller. The image features trajectory in the image plane is presented in Fig. 4b, which shows that both controllers can achieve the goal position (the plus, +, sign in the image) with a very small error. This figure shows also that the trajectory obtained with the inverse fuzzy model controller is smoother. This fact is due to the high proportional gain that the classical controller must have to follow the trajectory, which demands higher velocities. This effect is easily removed by slowing down the classical controller. But in this case, the fuzzy controller clearly outperforms the classical controller. The classical controller can only follow the trajectory without oscillations in the joint velocities if the robot takes 1.5s to move from one point to the other. In this case, the classical controller is about 50% slower than the fuzzy model-based controller proposed in this paper.

5 Conclusions

This paper introduces an eye-in-hand path planning image-based visual servoing scheme based on fuzzy modeling and control. The fuzzy modeling approach was applied to obtain an inverse model of the mapping between image features velocities and joints velocities. This inverse model is directly used as controller of a robotic manipulator performing path planning visual servoing, for a given image features velocity profile. The obtained results showed that both the classical and the fuzzy controllers can follow the image features velocity profile. However, the proportional gain of the classic visual servoing must be very high. This fact justifies the more smooth trajectory in the image plane for the fuzzy controller. For that reason, the inverse fuzzy control proposed in this paper performs better. As future work, the proposed fuzzy model based control scheme will be implemented in the experimental test-bed. Note that an off-line identification of the inverse fuzzy model must first be performed. The complete automation of this identification step is also under study.

Acknowledgement

This work is supported by the "Programa de Financiamento Plurianual de Unidades de I&D (POCTI) do Quadro Comunitário de Apoio III", by program FEDER, by the FCT project POCTI/ EME/39946/2001, and by the "Programa do FSE-UE, PRODEP III, acção 5.3, no âmbito do III Quadro Comunitário de apoio".

References

1. R. Babuška. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, 1998.
2. D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1/2):123–141, 1995.
3. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.
4. P. S. Gonçalves and J. C. Pinto. Camera configurations of a visual servoing setup, for a 2 dof planar robot. In *Proceedings of the 7th International IFAC Symposium on Robot Control, Wroclaw, Poland.*, pages 181–187, Wroclaw, Poland, Sept. 2003.
5. D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings IEEE CDC*, pages 761–766, San Diego, USA, 1979.
6. S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, 1996.
7. Y. Mezouar and F. Chaumette. Path planning for robust image-based control. *IEEE Transactions on Robotics and Automation*, 18(4):534–549, 2002.
8. J. Sousa, C. Silva, and J. S. da Costa. Fuzzy active noise modeling and control. *International Journal of Approximate Reasoning*, 33:51–70, April 2003.
9. I. Suh and T. Kim. Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators. *IEEE Trans. on Fuzzy Systems*, 2(3):203–220, 1994.
10. O. Tahri and F. Chaumette. Application of moment invariants to visual servoing. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 4276–4281, Taipeh, Taiwan, 2003.