Evolving Inverse Fuzzy Models for Uncalibrated Visual Servoing in 3D Workspace

P. J. S. Gonçalves[1,2] J.M.C. Sousa[2] J.R. Caldas Pinto[2]

1.Instituto Politécnico de Castelo Branco, Escola Superior de Tecnologia
Av. Empresário, 6000-767 Castelo Branco, Portugal
2.IDMEC/IST,Technical University of Lisbon (TU Lisbon)
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Email: pgoncalves@est.ipcb.pt, {j.sousa,jcpinto}@dem.ist.utl.pt

Abstract— *In this paper, evolving inverse fuzzy models obtained online for uncalibrated visual servoing in 3D workspace, are developed and validated in a six degrees of freedom robotic manipulator. This approach will recursively update the inverse fuzzy model based only on measurements at a given time instant. The uncalibrated approach does not require calibrated kinematic and camera models, as needed in classical visual servoing to obtain the Jacobian. Experimental results obtained in a PUMA robot performing eye-to-hand visual servoing in 3D workspace are used to demonstrate the validity of the proposed approach, when compared to the previous developed off-line learning.*

Keywords— Visual Servoing, Fuzzy Control, Robotics, Evolving Fuzzy Systems.

1 Introduction

Industrial robotic manipulators use sensor-based control to perform tasks, either in structured or unstructured environments. Vision sensors provide a vast amount of information on the environment in which robots move. Thus, vision is essential for robots working in unstructured environments. This type of sensors definitely enlarges the potential applications of the actual robotic manipulators. Visual servoing can be defined as a method to control dynamic systems using the information provided by visual sensors. In this paper, the control of a robot manipulator end-effector using a single camera looking to the robot (eye-to-hand) is addressed [1].

Early approaches for visual servoing are based on a model already known, i.e. the robot-camera model, from which the relation between the features and the robot kinematics is analytically obtained [2]. Apart from the stated approaches where the robot-camera model is already known, it can also be estimated [3, 2]. This type of systems, *Uncalibrated Visual Servoing*, can deal with unknown robot kinematics and/or unknown camera parameters. By using this type of robot-camera models, the system becomes independent of robot type, camera type or even camera location.

In this paper, the robot-camera model estimation, from motions in the 3D workspace, by learning is addressed both online and off-line using fuzzy techniques to obtain a controller capable of controlling the system. An inverse fuzzy model is used to derive the inverse robot-camera model, i.e. the Jacobian, in order to compute the joints and end-effector velocities in a straightforward manner. The inverse fuzzy model cab be applied directly as a controller, which is a simple way to implement a controller in real-time. Note that this feature is very important in robotic systems.

From the modeling techniques based on soft computing, fuzzy modeling is one of the most appealing. As the robotic manipulator together with the visual system is a nonlinear system, which is only partly known, it is advantageous to use fuzzy modeling as a way to derive a model (or an inverse model as in this case) based only on measurements. In this paper will be applied and compared two approaches, off-line [2], and on-line [4, 5]. The first approach have already proven to perform well in visual servoing systems. It's main drawback is that the learning must be performed off-line. Since the environment in which the robot operates can vary importantly, new models should be derived very often, which take time and also the robot must stop the task that is performing. That is why an on-line modeling of the visual servoing system is preferable. This approach was not yet applied to control robotic manipulators using vision and is the objective of the present paper.

The paper is organized as follows. Section 2 describes briefly the concept of visual servoing and presents the uncalibrated visual servoing approach. Section 3 presents very briefly on-line and off-line fuzzy modeling, and discusses the problem of identifying inverse fuzzy models directly from data measurements. Section 4 describes the experimental setup and presents the obtained results, where the identified inverse fuzzy models are discussed. Finally, Section 5 presents the conclusions and the future work.

2 Visual Servoing

Machine Vision and Robotics can be used together to control a robotic manipulator. This type of control, which is defined as Visual Servoing, uses visual information from the work environment to control a robotic manipulator performing a given task. In the following sub-sections classical and uncalibrated visual servoing are presented, along with their main goals.

2.1 Classical Visual Servoing

Within the visual servoing framework, there are three main categories related with the information obtained from the image features:

- *image-based visual servoing* [1, 6], which uses direct information from the object in the image, i.e. image features,

ISBN: 978-989-95079-6-8

- *position-based visual servoing* [1], which uses 3D information of the object from the image(s), e.g. the 3D coordinates of the object or the rotation and translation between the camera and object frames, obtained using a CAD model of the object to perform an on-line pose estimation,
- *hybrid visual servoing* [7], which combines the first two approaches and is a possible solution to some drawbacks of image-based and position-based methods [8].

In the following, the classical approaches to modeling and control of position-based visual servoing are presented.

2.1.1 Modeling the Position-Based Visual Servoing System

In this paper position-based visual servoing with 3D features [9] is used in an eye-to-hand system [1], where the camera is fixed and looking the robot and object. The 3D image features, s are 3D points of the object in the camera frame, p. The kinematic modeling of the transformation between the image features velocities, $\dot{s}$, and the joints velocities $\dot{q}$ is defined as follows [9]:

$$\dot{s} = \begin{bmatrix} -I_3 & S(p) \end{bmatrix} \cdot {}^{c}W_e \cdot {}^{e}J_R \cdot \dot{q} = J \cdot \dot{q}\,, \qquad (1)$$

where I_3 is the 3×3 identity matrix, $S(p)$ is the skew-symmetric matrix of the 3D point p, ${}^{c}W_e$ is defined as the transformation between the camera and end-effector frames velocities, and ${}^{e}J_R$ is the robot Jacobian. The 3D point is obtained from the captured image using a pose estimation algorithm [9].

2.2 Uncalibrated Visual Servoing

To derive an accurate Jacobian, J, a perfect modeling of the camera, the chosen image features, the position of the camera related to the world, 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. Remind that the Jacobian must be inverted to send the camera velocity to the robot inner control loop. When the Jacobian is singular, the control cannot be correctly performed.

There are visual servoing systems that obviate the calibration step and estimate the robot-camera model either online or offline. The robot-camera model may be estimated:

- analytically, using nonlinear least square optimization [10], and
- by learning or training, using fuzzy membership functions and neural networks [11, 2].

In addition, the control system may estimate an image Jacobian and use the known robot model, or a coupled robot-camera Jacobian may be estimated.

To overcome the difficulties regarding the Jacobian, a new type of differential relationship between the features and camera velocities was proposed in [11]. This approach estimates the variation of the image features, when an increment in the camera position is given, by using a relation G. This relation is divided into G_1 which relates the position of the camera and the image features, and F which relates their respective variation:

$$s + \delta s = G(q + \delta\, q) = G_1(q) + F(q, \delta\, q). \qquad (2)$$

Considering only the variations in (2):

$$\delta s = F(q, \delta\, q), \qquad (3)$$

the inverse function F^{-1} is given by:

$$\delta q = F^{-1}(\delta s, q)\,, \qquad (4)$$

and it states that the joint variation depends on the image features variation and the previous position of the robot manipulator. Equation (4) can be discretized as

$$\delta q(k) = F_k^{-1}(\delta s(k+1), q(k)). \qquad (5)$$

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 error, $\delta s(k+1)$, 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 proposed in this paper and presented in Section 3. This new approach to image-based visual servoing allows to overcome the problems stated previously regarding the Jacobian and the calibration of the robot-camera model.

3 Inverse Fuzzy Modeling

3.1 Off-Line Fuzzy Modeling

Fuzzy modeling often follows the approach of encoding expert knowledge expressed in a verbal form in a collection of if–then rules. Parameters in this structure can be adapted using input-output data. When no prior knowledge about the system is available, a fuzzy model can be constructed entirely on the basis of system measurements. In the following, we consider data-driven modeling based on fuzzy clustering [12, 13].

We consider rule-based models of the Takagi-Sugeno (TS) type. TS models consist of fuzzy rules describing a local input-output relation, typically in an affine form:

$$\begin{aligned} R_i : \textbf{If } & x_1 \text{ is } A_{i1} \textbf{ and } \ldots \textbf{and } x_n \text{ is } A_{in} \\ \textbf{then } & y_i = \mathbf{a}_i \mathbf{x} + b_i\,, \quad i = 1, 2, \ldots, K. \end{aligned} \qquad (6)$$

Here R_i is the ith rule, $\mathbf{x} = [x_1, \ldots, x_n]^T$ are the antecedent variables, $A_{i1}, \ldots, A_{in}$ 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, and the aggregated output of the model, $\hat{y}$, is calculated by taking the weighted average of the rule consequents:

$$\hat{y} = \frac{\sum_{i=1}^{K} \beta_i y_i}{\sum_{i=1}^{K} \beta_i}\,, \qquad (7)$$

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 $A_{ij}(x_j) : \mathbb{R} \rightarrow [0, 1]$ is the membership function of the fuzzy set A_{ij} in the antecedent of R_i.

To identify the model in (6), 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

ISBN: 978-989-95079-6-8

in the product space of the inputs and the outputs [13]. Hence, the data set Z to be clustered is composed from X and $\mathbf{y}$: $\mathrm{Z}^T = [\mathrm{X}, \mathbf{y}]$. Given Z and an estimated number of clusters K, the Gustafson-Kessel fuzzy clustering algorithm [14] is applied to compute the fuzzy partition matrix U.

The fuzzy sets in the antecedent of the rules 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. One-dimensional fuzzy sets A_{ij} are obtained from the multi-dimensional fuzzy sets defined point-wise in the ith row of the partition matrix by projections onto the space of the input variables x_j. The point-wise defined fuzzy sets A_{ij} are approximated by suitable parametric functions in order to compute $\mu_{A_{ij}}(x_j)$ for any value of x_j.

The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let $\theta_i^T = [\mathbf{a}_i^T; b_i]$, 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 $\mathbf{y} = \mathrm{X}_e\theta + \epsilon$ becomes

$$\theta_i = \left[\mathrm{X}_e^T \mathrm{W}_i \mathrm{X}_e\right]^{-1} \mathrm{X}_e^T \mathrm{W}_i \mathbf{y}\,. \tag{8}$$

3.2 On-Line Fuzzy Modeling

The model obtained from the techniques presented in the previous section is assumed to be fixed, since it is learned in off-line mode. Recently attention is focused in on-line learning [4], where in a first phase input-output data is partitioned using unsupervised clustering methods and in a second phase, parameter identification is performed using a supervised learning method.

In On-Line Fuzzy Modeling and according to [4], also rule-based models of the TS type, are considered. Typically in the affine form described in (6), where the input-output data is acquired continuously. The new data, arriving at some time instant, can bring new information from the system, which could indicate a change in its dynamics. This information may change an existing rule, by changing the spread of the membership functions, or even introduce a new one. To achieve this, the algorithm must be able to judge the informative potential and the importance of the new data.

In the following are briefly presented the several steps of the algorithm used for on-line fuzzy modeling, proposed in [4], evolving fuzzy systems. The first step is based on the subtractive clustering algorithm [15], where the input-output data is partitioned. The procedure used must be initialized, i.e. the focal point of the first cluster is equal to the first data point and its potential is equal to one. Starting from the first data point, the potential of the next data point is calculated recursively using a Cauchy type function of first order:

$$P_k(z_k) = \frac{1}{1 + \frac{1}{k-1}\sum_{i=1}^{k-1}\sum_{j=1}^{n+1}(d_{ik}^j)^2}, \quad k = 2, 3, \ldots \tag{9}$$

where $P_k(z_k)$ denotes the potential of the data point z_k calculated at time k; $d_{ik}^j = z_i^j - z_k^j$, denotes projection of the distance between two data points (z_i^j and z_k^j) on the axis z^j.

When a new data point arrives it also influences the potential of the already defined center of the clusters ($z_i^*, i = 1, 2, ..., K$). A recursive formula for the update of the cluster centers potential is defined in [4]:

$$P_k(z_i^*) = \frac{(k-1)P_{(k-1)}(z_i^*)}{k - 2 + P_{(k-1)}(z_i^*) + P_{(k-1)}(z_i^*) + \sum_{j=1}^{n+1}(d_{ik}^j)^2},$$

where $P_k(z_i^*)$ is the potential at time k of the cluster center, related to the rule i.

The next step of the algorithm is to compare the potential of the actual data point to the potentials of the existing cluster centers. According to the two approaches tested in this paper [4, 5] a crucial part of the evolving fuzzy systems is the rule creation and modification. The first approach is called Evolving Takagi-Sugeno (eTS) and the second eXtended Takagi-Sugeno (xTS).

In eTS, if the potential of a new data point is higher than the potential of the existing cluster centers, then the new data point is accepted as a new cluster center and a new rule is formed. If in addition to the previous condition the new data point is close to an old cluster center, the old cluster center is replaced. This last condition is defined by:

$$\frac{P_k(Z_k)}{\underset{i=1}{\overset{K}{max}} P_k(z_i^*)} - \frac{\delta_{min}}{r} \geq 1, \tag{10}$$

where $r \in [0.3; 0.5]$ is the spread of the antecedent [4], and δ_{min} is the shortest distance between the new candidate Z_k and all the existing cluster centers z_i^*.

In xTS, if the potential of a new data point is higher than the maximum potential of all the existing clusters centers or is lower than the minimum potential of all the existing clusters centers, then the new data point is accepted as a new cluster center and a new rule is formed. In addition to the previous condition, if the next condition is true then the old cluster is replaced because the candidate is close to an old cluster center:

$$\underset{i=1}{\overset{K}{min}} \|z_k - z_i^*\|_j \geq \frac{r_j^i}{2}, \tag{11}$$

where r_j^i is the spread that is not fixed, like in the previous approach. In this approach the spread is adaptive and can vary in each element j of every cluster (rule) i, making possible to obtain hyper-ellipsoidal clusters. The recursive formula to obtain the adaptive spread is as follows:

$$r_{jk}^p = \rho \cdot r_{j(k-1)}^p + 0.5\sqrt{D_{jk}^p}; p = \underset{i=1}{\overset{K}{argmin}} \|z_k - z_i^*\|, \tag{12}$$

where D_{jk}^p is the local scatter and $\rho = 0.75$, both defined in [5].

For both approaches the fuzzy sets in the antecedent of the rules are gaussian, with the form:

$$\mu_{ij} = e^{-r\|x_j - x_{ij}^*\|^2}, \tag{13}$$

The consequents of the fuzzy rules are obtained using the global parameter estimation procedure based on the weighted recursive least squares, presented in [4].

3.3 Inverse modeling

For the robotic application in this paper, the inverse model is identified using input-output data from the inputs $\dot{q}(k)$ and

ISBN: 978-989-95079-6-8

Figure 1: The eye-to-hand experimental setup.

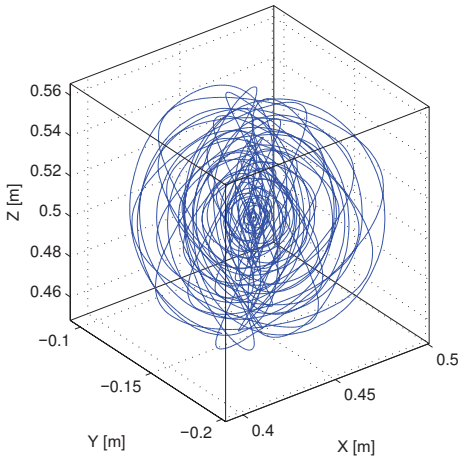


Figure 2: Robot end-effector 3D spiral trajectory.

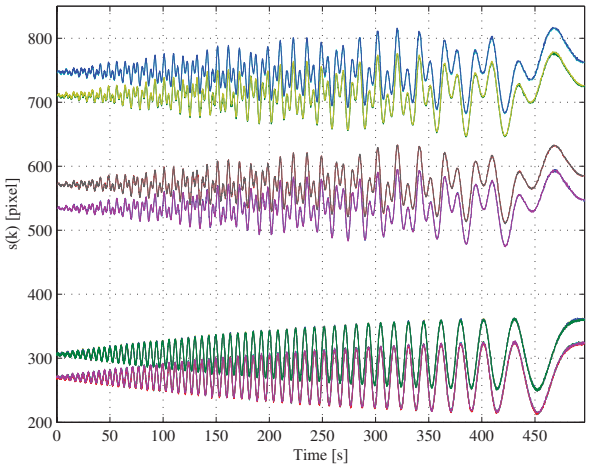


Figure 3: Input data for inverse model identification, image features $s(k)$.

the outputs $\delta s(k+1)$, following the procedure described in [2]. In this paper, the approach presented in [3] to obtain the 3D training set, was used. Note that we are interested in the identification of an inverse model as in (5).

4 Results

4.1 Experimental Setup

A Puma 560 Robotic Manipulator and a Vector CCi4 camera, in eye-to-hand configuration, were used to demonstrate the validity of the approach proposed in this paper. The experimental setup is presented in Fig. 1. The visual control algorithms were implemented in real-time using MatLab Simulink, and the xPC Target toolbox. The robot inner loop velocity control, performs at 1 KHz and the visual loop control at 12.5 Hz. A planar object is rigidly attached to the robot end-effector and is described with eight points, which centroids were used as the sixteen image features. In this paper, the robot moves in its 3D workspace (moving joints 1, 2 and 3). To maintain the image features in the camera field of view during the servoing, the planar target is set to be perpendicular to the camera optical axis by moving joints 4 and 5, as shown in Fig. 1.

4.2 Inverse Modeling Results

Following the work in [3], to obtain the identification data, the robot swept the 3D workspace in the camera field of view in a 3D spiral path, starting in the spiral center, Fig. 2, which allows to obtain the model for the end-effector position.

The variables needed for identification, $\delta q(k)$ and $\delta s(k+1)$, are obtained from the spiral when setting the desired position to the spirals center, by:

$$\delta q(k) = \frac{q^* - q(k)}{\Delta t} \tag{14}$$

$$\delta s(k+1) = s^* - s(k+1) \tag{15}$$

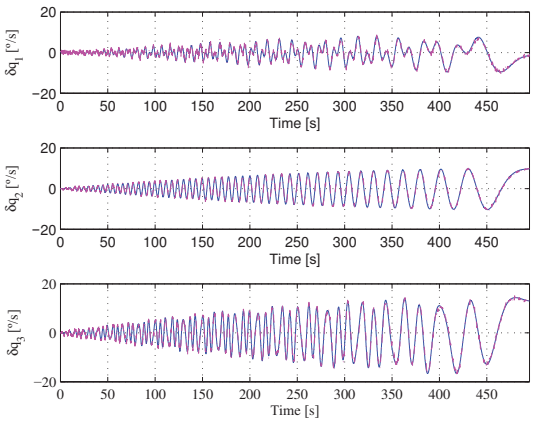
This allows to cover a wide range of values for $\delta q(k)$ and $\delta s(k+1)$. The 3D spiral also allows to control the precision of the model, by increasing or decreasing its parameters. Moving the first, the second and third joints of the PUMA Robotic Manipulator, the end-effector position follows the desired 3D spiral. The end-effector orientation is set to maintain the target parallel to the image plane.

The inverse fuzzy model was identified from the 3D spiral trajectory data, i.e. the joint velocities (see Fig. 4) and the sixteen image features obtained (see Fig. 3) using (14).

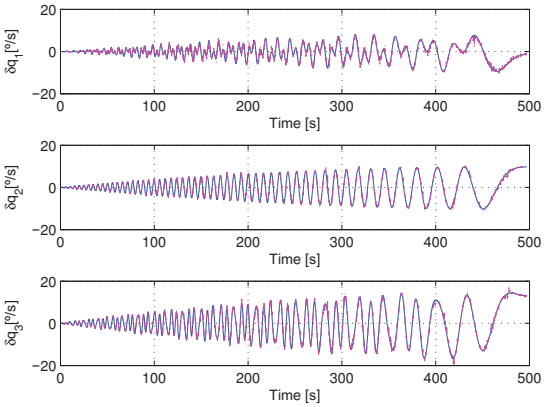
Note that to identify the inverse model, one cannot simply feed the inputs as outputs and the outputs as inputs. Since the inverse model (5) is a non-causal model, the output of the inverse model must be shifted one step, see [16].

For the identification of joints 1, 2 and 3, the obtained joint velocities $\dot{q}(k)$, are shown in Fig. 4, solid lines. Note that three fuzzy models are identified, one for each joint velocity. In the plots it is hard to see the difference between the real output data and the output of the inverse fuzzy model, because they are very similar, output of the models in dash-dotted lines. In order to measure modeling accuracy, this paper uses the Variance Accounted For (VAF) [2].

ISBN: 978-989-95079-6-8



(a) Comparison of the output data of the off-line inverse fuzzy model, $\dot{q}(k)$, joints 1 to 3. Solid – real output data, and dash-dotted – output of the off-line inverse fuzzy model.



(b) Comparison of the output data of the on-line inverse fuzzy model (xTs), $\dot{q}(k)$, joints 1 to 3. Solid – real output data, and dash-dotted – output of the on-line inverse fuzzy model.

Figure 4: Output data for inverse model identification, joints 1 to 3.

In Table 1 are presented the VAF's for the off-line modeling (oTS) and for the two on-line modeling approaches (eTS and xTS). The best result is for the off-line case as expected, although the on-line case also performs very well. The best result for on-line modeling is presented by xTS but it costs more rules, 110, and consequently more computational time to update all the cluster centers and rules.

Note that the number of clusters is pre-defined to four in oTS and that the spread $r = 0.4$ for eTS. In xTS the spread is adaptive.

5 Conclusions and Future Works

This paper introduces a novel contribution to eye-to-hand visual servoing, based on on-line fuzzy modeling to obtain an uncalibrated visual servoing system. Two methods for on-line fuzzy modeling (eTS) and (xTS) have showed excellent results when compared with the results obtained for the off-line approach. This proves the validity of the proposed approach

Table 1: Results of the inverse fuzzy model, obtained for each joint with the three different approaches.

	Joint 1		Joint 2		Joint 3	
	VAF	Rules	VAF	Rules	VAF	Rules
oTS	96.4%	4	98.5%	4	98.5%	4
eTS	95.6%	50	98.4%	50	98.5%	50
xTS	95.6%	110	98.4%	110	98.6%	110

to uncalibrated visual servoing. With the robot-camera model being able to adapt on-line to changes in the environment, the robot can now operate on different conditions without the need for re-learning off-line the model or perform any kind of calibration.

As expected the off-line modeling produces better results when compared to both on-line approaches, because it takes into account all the data points. In on-line modeling, only the actual data point and the defined cluster centers are used to update the model, introducing or replacing rules. That is why the on-line approach performs not as good as the off-line approach. However we are convicted that the differences in the models, will not compromise the use of the model to control the robot.

As future work, the proposed evolving inverse fuzzy model approach will be extended to the tri-dimensional robot workspace and also will be tested to control the PUMA Robotic Manipulator for the straight line trajectory presented in Fig. 2, in order to verify the results already achieved for the inverse model obtained off-line. The influence of using the eTS or xTS models will also be studied, i.e. the trade-off between accuracy (VAF) and computational effort (number of rules), when controlling the robot.

Acknowledgements

This work was funded by FCT trough "Programa POCI2010, Unidade 46", subsidized by FEDER.

References

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

[2] P.J. Sequeira Gonçalves, L.F. Mendonça, J.M. Sousa, and J.R. Caldas Pinto. Uncalibrated eye-to-hand visual servoing using inverse fuzzy models. *IEEE Transactions on Fuzzy Systems*, 16(2):341–353, 2008.

[3] P.J. Sequeira Gonçalves, A. Paris, C. Christo, J.M.C. Sousa, and J.R. Caldas Pinto. Uncalibrated visual servoing in 3d workspace. *Lecture Notes in Computer Science*, 4142:225–236, 2006.

[4] P. Angelov and D. Filev. An approach to online identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 34:484–498, 2004.

[5] P. Angelov and Xiaowei Zhou. Evolving fuzzy systems from data streams in real-time. In *Proceedings of the International Symposium on Evolving Fuzzy Systems*, pages 29–35, Ambelside, Lake District, UK, 2006.

[6] 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.

ISBN: 978-989-95079-6-8

[7] E. Malis, F. Chaumette, and S. Boudet. 2 $\frac{1}{2}$ d visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, 1999.

[8] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The Confluence of Vision and Control, D. Kriegman, G . Hager, A.S. Morse (eds.), LNCIS Series*, (1998):66–78, 2000.

[9] E. Cervera and P. Martinet. Combining pixel and depth information in image-based visual servoing. In *Proceedings of the Ninth International Conference on Advanced Robotics*, pages 445–450, Tokyo, Japan, 1999.

[10] J. Peipmeier, G. McMurray, and H. Lipkin. Uncalibrated dynamic visual servoing. *IEEE Trans. on Robotics and Automation*, 20(1):143–147, February 2004.

[11] I.H. Suh and T.W. Kim. Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators. *IEEE Transactions on Fuzzy Systems*, 2(3):203–220, 1994.

[12] R. Babuška. *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston, 1998.

[13] J.M. Sousa and U. Kaymak. *Fuzzy Decision Making in Modeling and Control*. World Scientific Pub. Co., Singapore, 2002.

[14] 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.

[15] S. L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent Fuzzy Systems*, 2:267–278, 1994.

[16] J.M. Sousa, C. Silva, and J. Sá da Costa. Fuzzy active noise modeling and control. *International Journal of Approximate Reasoning*, 33:51–70, April 2003.

ISBN: 978-989-95079-6-8