Learning Approaches to Visual Control of Robotic Manipulators

Paulo J. S. Gonçalves*†, Pedro M. B. Torres*
*School of Technology,
Polytechnic Institute of Castelo Branco
Av. Empresário, 6000-767 Castelo Branco, Portugal
Email: pedrotorres@ipcb.pt
†Technical University of Lisbon, IDMEC / IST,
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
Email: paulo.goncalves@ipcb.pt

***Abstract*—This paper presents learning approaches to model the interaction between a robotic manipulator and its working environment. The approaches used are fuzzy modeling, neural networks and support vector machines. The interaction tackled in this paper is between the robot visual perception of the work environment and its actuators, while performing positioning tasks. This interaction, e.g., model, is obtaining only on measurements. This fact allows to obtain an uncalibrated model of the interaction, minimizing the setup time of the robotic system, not requiring calibrated robot kinematic and camera models. The input-output sample data used to learn the model are visual features from the work environment and the robot joint velocities, respectively. Experimental data, obtained from a IR52C robot and a visual stereo system, was used to validate the obtained models. Due to its accuracy and lower computational complexity, when compared to the other three, the off-line fuzzy model was used to control the robot, which clearly shows the effectiveness of the approach.**

***Keywords*-Fuzzy Modeling, Neural Networks, Support Vector Machines, Computer Vision, Robotic Manipulators.**

I. Introduction

Visual servo control of robots is an area in continuous expansion, since vision sensors provide much more information about the working environment of the robot, than any other type of sensors. The area of robotics that addresses this concept is also called Visual Servoing, which essentially defines the control methods for dynamic systems, using information from vision sensors (cameras). The basis of this work came from the idea of modeling the interaction between the motion and vision of an industrial robot manipulator, without a-priori calibration of the system. Industrial manipulators are usually programmed using 3D coordinates of the work environment. This lead us to use stereo vision to obtain the image features, in order to obtain the 3D coordinates to control the robot.

To perform Uncalibrated Visual Servo Control, the robot-camera model must be estimated. Previous work from the authors [5], showed that Uncalibrated Visual Servo Control can be applied to control an industrial manipulator using vision, with the following advantages: no need to calibrate the robot; no need to calibrate the camera(s); the controller has no singularities. In [5] was developed a system based on off-line fuzzy modeling.

In this paper, the estimation is performed by learning. Four approaches are presented, the first off-line fuzzy modeling [5], the second on-line fuzzy modeling [11], the third neural networks [13] and the fourth support vector machines [14]. Neural networks and support vector machines are two major machine learning approaches, and were not yet applied to the estimation of the robot-camera model. These approaches are used to compare the previous work results [5], obtained using the off-line fuzzy modeling, and also to find a better alternative to fuzzy modeling. On-line fuzzy modeling pretend to be an extension of the off-line fuzzy modeling approach. These four approaches lead to a model capable of controlling the visual servo system. The learning approaches are used to derive the inverse robot-camera model, i.e., the inverse Jacobian, in order to compute the joints and end-effector velocities in a straightforward manner. The models can be directly applied 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.

This paper is organized as follows. Section II describes briefly the concept of visual servo control and presents the uncalibrated visual servo control approach. Section III presents very briefly on-line and off-line fuzzy modeling, neural networks and support vector machines. Section IV describes the experimental setup and presents the obtained results, where the identified models are discussed. Finally, Section V presents the conclusions and the future work.

II. Visual Servo Control

A. Calibrated Visual Servo Control

In this paper 3D visual servoing with 3D features, [1], is used in an eye-to-hand system, [2], where the camera is fixed and looking the robot and the 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 $\dot{q}$

 ISBN: 978-1-61208-108-3

is defined as follows, [1]:

$$\dot{s} = [\ -I_3 \quad S(p)\ ] \cdot {}^{c}W_e \cdot {}^{e}J_R \cdot \dot{q} = J \cdot \dot{q}\,, \qquad (1)$$

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, [1].

B. Uncalibrated Visual Servo Control

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 servo control 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 [3], and
- By learning or training, using fuzzy membership functions and neural networks [4], [5].

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 differential relationship between the features and camera velocities was proposed in [4]. This approach states that the joint variation depends on the image features variation and the previous position of the robot manipulator:

$$\delta q(k) = F_k^{-1}(\delta s(k+1), q(k)). \qquad (2)$$

In visual servo control, 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 fuzzy modeling as proposed in this paper and presented in Section III. This new approach to visual servo control allows to overcome the problems stated previously regarding the Jacobian and the calibration of the robot-camera model. It can be applied to all types of visual servo control. It was apllied to 2D in [5], and it will be applied to 3D in this paper.

III. Learning Approaches to Modeling

A. Fuzzy Models

From the modeling techniques based on soft computing, fuzzy modeling is one of the most appealing. If no a priori knowledge is available, the rules and membership functions can be directly extracted from process measurement. Fuzzy models provide a transparent description of the system, that can reflect a possible nonlinearity of the system. The fuzzy models implemented in the presented toolbox are Takagi-Sugeno fuzzy models [6] where the consequents are crisp functions of the antecedent variables and linguistic or Mandani [7], [8] fuzzy models where both the antecedent and consequent are fuzzy propositions.

1) Takagi Sugeno: Takagi-Sugeno (TS) models consist of fuzzy rules describing a local input-output relation, typically in an affine form:

$$\begin{aligned} R_i : &\ \textbf{If } x_1 \text{ is } A_{i1} \textbf{ and } \ldots \textbf{and } x_n \text{ is } A_{in} \\ &\ \textbf{then } y_i = \mathbf{a}_i \mathbf{x} + b_i\,, \quad i = 1, 2, \ldots, K. \end{aligned} \qquad (3)$$

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:

$$\hat{y} = \frac{\sum_{i=1}^{K} \beta_i y_i}{\sum_{i=1}^{K} \beta_i}\,, \qquad (4)$$

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

To identify the model in (3), 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 [9]. 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 [10] 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 multidimensional 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 $\theta_i^T = \left[\mathbf{a}_i^T; b_i\right]$, 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 $\mathbf{y} = \mathrm{X}_e\theta + \epsilon$ becomes

$$\theta_i = \left[\mathrm{X}_e^T \mathrm{W}_i \mathrm{X}_e\right]^{-1} \mathrm{X}_e^T \mathrm{W}_i \mathbf{y}\,. \tag{5}$$

Previous work of the first author have already stated that this approach can obtain a model capable of controlling an image based visual servoing system [5].

2) Evolving: The model obtained from the techniques presented in the previous two sections is assumed to be fixed, since it is learned in off-line mode. Recently attention is focused in on-line learning [11], 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 [11], also rule-based models of the TS type, are considered. Typically in the affine form described in (3), 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 is briefly presented the on-line fuzzy modeling algorithm, proposed in [11], called evolving fuzzy systems. The first step is based on the subtractive clustering algorithm [12], 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:

$$P_k(z_k) = \frac{1}{1 + \frac{1}{k-1}\sum_{i=1}^{k-1}\sum_{j=1}^{n+1}(d_{ik}^j)^2}, \quad k = 2, 3, ... \tag{6}$$

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 K clusters $(z_i^*, i = 1, 2, ..., K)$. A recursive formula for the update of the cluster centers potential is defined in [11]:

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

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.

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. The decision to create or remove rules was based on the following principles:

1)The sample has a high potential is legible to be a focal point of a fuzzy rule:

$$P_k(z_k) > \max(P_k(z_i^*)) \tag{7}$$

2)A sample that is over an area of spatial data are is not covered by other rules, is also eligible to form a rule:

$$P_k(z_k) < \min(P_k(z_i^*)) \tag{8}$$

3)To avoid overlap and redundancy of information in creating new rules, the following condition is also checked:

$$\exists i, i = [1, R]; \mu_{ij}(x(k)) > e^{-1}; \forall j; j = [1, n] \tag{9}$$

R denotes the number of fuzzy rules up to the moment k. The membership function are gaussian, with the form:

$$\mu_{ij} = e^{-r\|x_j - x_{ij}^*\|^2}, \tag{10}$$

The consequents of the fuzzy rules are obtained using the global parameter estimation procedure based on the weighted recursive least squares, presented in [11].

B. Neural Networks

Neural networks have found profound success in the area of pattern recognition, function approximation, optimization, pattern matching and associative memories. By repeatedly showing a neural network inputs classified into groups, the network can be trained to discern the criteria used to classify, and it can do so in a generalized manner allowing successful classification of new inputs not used during training [13]. In the present paper a feedforward backpropagation network with 5 neurons, with sigmoid activation functions in the hidden layer and a linear one in the output layer, is used to obtain the model of the interaction.

C. Support Vector Machines

The support vector machine (SVM) maps an input vector x into a high-dimensional feature space Z through some nonlinear mapping, chosen a priori. In this space, optimal separating hyperplanes are constructed. In the case of regression, SVM performs modeling between several clusters by finding decision hyper surfaces determined by certain points of the training set, termed Support Vectors [14].

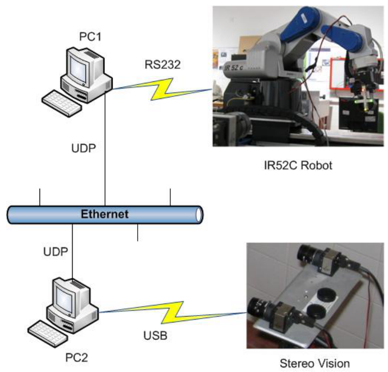

Figure 1. The eye-to-hand experimental setup.

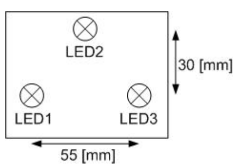

Figure 2. Configuration of the markers placed on the end-effector.

IV. Experimental Results

A. Experimental Setup

To validate the proposed approach, we use the IR52C Robotic Manipulator and a stereo vision system, with *U-Eye* cameras, in eye-to-hand configuration. The experimental setup is presented in Fig. 1. The PC2 acquire and process [15] the images from cameras and sends 3D data to the network using UDP protocol. Computer PC1 receives UDP packets coming from the PC2, and implement the visual servo control algorithm, to control the robot. To extract the 3D features for visual servo control, an object with tree LEDs, was placed in end-effector, with the marker configuration depicted in Figure 2. Computer PC2 acquires images from cameras, perform color segmentation of images, and extract the 3D coordinates, in real-time, corresponding to the 3D position of the three markers, [15]. These nine features are send to the network.

1) Modeling Results: For the robotic application in this paper, the models are identified using input-output data from the inputs $\dot{q}(k)$ and the outputs $\delta s(k+1)$, following the procedure described in [5]. In this paper, the approach presented in [16] to obtain the training set was used. Note that we are interested in the identification of an inverse model as in equation 2.

To obtain the data for model identification, the robot must move in the 3D workspace within the field of view of cameras, making a 3D spiral with a center point, (Fig. 3). The variables needed for identification, $\dot{q}(k)$ and $\delta s(k+1)$, are obtained from the spiral. This allows to cover a wide range of values for $\dot{q}(k)$ and $\delta s(k+1)$, by the equations 11 and 12.

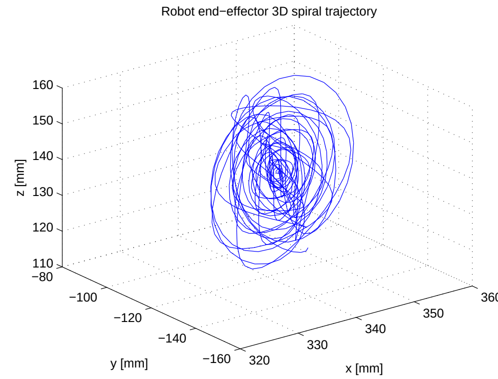

Figure 3. 3D Spiral path for model identification.

Table I
RESULTS OF THE OFF-LINE FUZZY MODEL

	Rules	VAF	MSE
Joint 1	3	98,2%	0,23
Joint 2	3	97,3%	0,93
Joint 3	3	94,4%	2,81
Joint 4	3	93,2%	1,21
Joint 5	3	98,2%	0,23

$$\delta s(k+1) = s^* - s(k+1) \tag{11}$$

$$\dot{q}(k) = \frac{q^* - q(k)}{\Delta t} \tag{12}$$

To estimate the modeling accuracy, we use the VAF (Variance Accounted For), defined in equation 13, where "cov" represent the covariance vector, and "MSE" (Mean Squared Error), defined in equation 14. A perfect match occurs, when VAF is 100% and MSE have value 0.

$$VAF = 1 - \frac{cov(y_i - \widehat{y}_i)}{cov(y_i)} \times 100\% \tag{13}$$

$$MSE = \frac{1}{n}\sum(\widehat{y}_i - y_i)^2 \tag{14}$$

In Table 1, are presented the values of VAF and MSE for the off-line fuzzy modeling. With only three rules, excellent values of VAF and MSE were obtained, meaning that the model is good for estimating the joint velocities.

In Off-Line Fuzzy Modeling the number of clusters (rules) must be defined a priori in order to obtain a model. In On-Line Fuzzy Modeling, evolutionary algorithms are used after initialization, and will estimate the number of rules required in accordance with the potential associated with each data. The results from On-Line Fuzzy Modeling are presented in Table 2. The variable Ω is the initialization parameter of the algorithm, that varies with the type of data.

Table II
RESULTS OF THE ON-LINE FUZZY MODEL

	Ω	VAF	MSE	Rules
Joint 1	400	97,9%	0,04	82
Joint 2	215	94,1%	0,37	74
Joint 3	250	95,6%	0,36	75
Joint 4	250	93,1%	0,20	66
Joint 5	342	64,3%	1,18	82

Table III
RESULTS OF THE NEURAL METHOD MODEL

	VAF	MSE
Joint 1	98,3%	0,22
Joint 2	97,3%	0,94
Joint 3	94,4%	2,84
Joint 4	93,9%	1,08
Joint 5	98,2%	0,23

The results presented in Table 2, show very good results with the exception of the last joint, but at the expense of a high number of rules, which will hopefully be minimized in future works. In Table 3 the results obtained with the neural network approach are presented, with similar results to the fuzzy off-line approach with only 5 neurons in the hidden layer. The SVM approach lead to better results, presented in Table 4, but at expenses of the complexity of the model since there are almost 900 support vectors. This fact does not allow a real time control of the robot, because of the computational time.

From the presented results of the four learning approaches, the on-line fuzzy model gives the best results, when taking only into account the error parameters, MSE or VAF. Since the model must be implemented to control the robot, the computational time is very important. Taking this parameter into account, the off-line fuzzy model must be used for control because it only has 3 if-then rules when compared to the 82 rules, only for joint 1, of the on-line model. The computational complexity of the neural networks and the SVM is similar to the on-line model case.

2) Control Results: The modeling results obtained lead us to implement the off-line approach because of the simplicity of the model (only 3 if-then rules), when compared to the neural network and the support vector machine. The on-line fuzzy model have not achieved good results for control, specially due to joint 5.

The control results were obtained using the Off-Line fuzzy model based control, defined in Fig. 4.

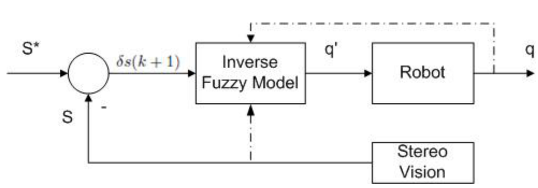

Figure 4. Uncalibrated Visual Servo Control Loop.

Table IV
RESULTS OF THE SUPPORT VECTOR MACHINE MODEL

	VAF	MSE
Joint 1	99,7%	0,04
Joint 2	99,6%	0,14
Joint 3	99,2%	0,39
Joint 4	99,3%	0,13
Joint 5	99,6%	0,04

To test the fuzzy models estimated, some trajectories were set within the workspace of the robot. The results obtained are quite satisfactory. Although some initial oscillations of the 3D positions error, the robot could stabilize and stop at a position close to the desired value of presenting a small error, within 3mm. Figure 5, Figure 6 and Figure 7, show the error for each marker, with respect to the 3D coordinates X, Y, Z, respectively, obtained in one of several trajectories performed with the robot. The control approach can stabilize the robot, as depicted in Figures 5 to 7, which shows the evolution of the robot joint velocities during the trajectory.

V. CONCLUSIONS AND FUTURE WORK

This paper presented a comparison between four learning approaches to obtain the interaction between the robot manipulator actuators and vision, when the robot performs positioning tasks. Four methods are presented and compared: Off-line Fuzzy Modeling, On-Line Fuzzy Modeling, Neural Networks and Support Vector Machines. The Off-line Fuzzy Modeling approach proven to be the adequate choice to control the robotic manipulator. With the Off-Line Model, was implemented a controller based on the learned fuzzy model, to control the IR52C robot to perform trajectories in its workspace. This controller presented very good results, with errors within 3mm of the desired position. The future goal is to implement a procedure that can update on-line the off-line learned model. For that, the first steps were already accomplished, i.e., a model based on Evolving Takagi-Sugeno Fuzzy Systems was obtained in this paper. With this approach very good results of VAF and MSE were obtained, with very satisfactory accuracy. The main objective of the future work is to reduce the number of rules of the on-line model, to allow that an On-Line Fuzzy model can control the IR52C Robotic Manipulator.

ACKNOWLEDGMENT

The authors would like to thank the Portuguese Science Foundation, FCT, for the funding to IDMEC through the Associated Laboratory in Energy, Transports, Aeronautics and Space (LAETA). To the FCT project: PTDC/EME-CRO/099333/2008.

REFERENCES

[1] 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*, Tokyo, Japan, 1999, pp. 445–450.

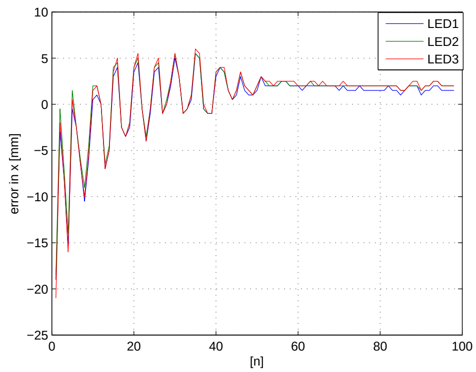

Figure 5. Evolution of the error of the position on the X coordinate.

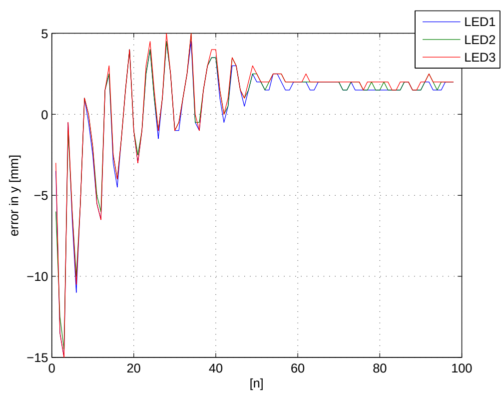

Figure 6. Evolution of the error of the position on the Y coordinate.

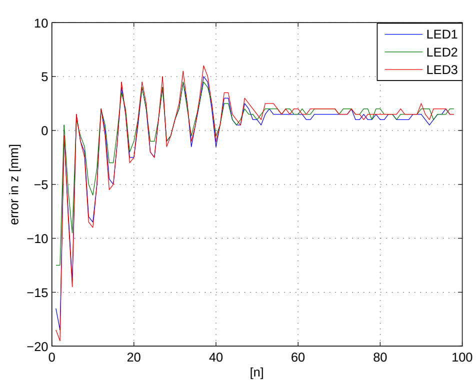

Figure 7. Evolution of the error of the position on the Z coordinate.

[2] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[3] J. Peipmeier, G. McMurray, and H. Lipkin, "Uncalibrated dynamic visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 20, no. 1, pp. 143–147, February 2004.

[4] I. Suh and T. Kim, "Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators," *IEEE Transactions on Fuzzy Systems*, vol. 2, no. 3, pp. 203–220, 1994.

[5] P. S. Gonçalves, L. Mendonça, J. Sousa, and J. C. Pinto, "Uncalibrated eye-to-hand visual servoing using inverse fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 341–353, 2008.

[6] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modelling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116–132, 1985.

[7] L. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 1, pp. 28–44, 1973.

[8] E. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic systems," *IEEE Transactions on Computers*, vol. 26, no. 12, pp. 1182–1191, 1977.

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

[10] D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proceedings IEEE CDC*, San Diego, USA, 1979, pp. 761–766.

[11] 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*, vol. 34, pp. 484–498, 2004.

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

[13] G. Zhang, "Neural networks for classification: a survey," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000.

[14] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

[15] P. Morgado, J. C. Pinto, J. M. M. Martins, and P. Gonçalves, "Cooperative eye-in-hand/stereo eye-to-hand visual servoing," in *Proc. of RecPad 2009 - 15th Portuguese Conference in Pattern Recognition*, Aveiro, Portugal, 2009.

[16] P. S. Gonçalves, A. Paris, C. Christo, J. Sousa, and J. C. Pinto, "Uncalibrated visual servoing in 3d workspace," *Lecture Notes in Computer Science*, vol. 4142, pp. 225–236, 2006.