Forward Model Creation for Visual Servoing in a Six Link Manipulator

Lord Kenneth Pinpin, Daniel Fernando Tello Gamarra, Cecilia Laschi and Paolo Dario

Abstract— **This paper shows the utility of a forward model for improving the visual servoing performance of a robotic manipulator. Following a developmental robotics approach, the manipulator undergoes a motor babbling phase which is used to create a forward model using an ANFIS neural network. The forward model maps the relationship between the joint positions and the image of the end effector in two camera views. Using the obtained forward model, an initial image Jacobian is estimated and is used with a visual servoing controller. Simulation results demonstrate that errors are significantly lower when the estimated Jacobian is used.**

I. INTRODUCTION

RECENT studies point to the possibility that human beings could create internal models [1]. Kawato in [2] defined internal models as neural mechanisms that can mimic the input/output characteristics, or their inverses of the motor apparatus. The internal models could be forward models or inverse models. Forward models can predict sensory consequences from efference copies of issued motor commands. Inverse models calculate the necessary feedforward motor commands from desired trajectory information.

The process of creation of forward models starts in infants when they born. The newborn through a self exploratory phase of his kinematics and sensory feedback ("body babbling") creates an internal model of his own kinematics and sensory system as described in [3]. Roboticians looking at biology, and specifically human development, as a source of inspiration have began to use forward models in robots. For instance, in [4] a forward model that represents the forward kinematics of a manipulator was created using Radial Basis function neural networks. From the forward model they derived analytically the robot Jacobian that is used in a control law that governs the reaching. In [5] a mobile robot, after a babbling motor phase, learns a forward model based on a Bayesian neural network. The forward model was used by the robot in imitating human movements.

Following a developmental robotics roadmap, this work tries to shed some light in the use of forward models for visual servoing that we intend to use for vision based reaching. The forward model created is used to estimate an initial image Jacobian that becomes an important factor that determines the maximum performance attainable in a reaching task using a well known visual servo controller.

The remainder of the paper is as follows. In the second section are described the algorithms used in this work; the third section shows the general architecture for the reaching task; in the fourth section the babbling motor phase is described; in the fifth section the forward model creation using an ANFIS is detailed; the sixth section presents the results of the controller performance in a reaching task, finally, conclusions and planned future works are described in the sixth section.

Manuscript received April 23, 2008. This work was supported by the ROBOTCUB project (IST-2004-004370), funded by the European commission through unit E5 "Cognitive Systems".

Lord Kenneth Pinpin, Daniel Fernando Tello Gamarra, Cecilia Laschi and Paolo Dario are with the Scuola Superiore Sant'Anna, ARTS Lab (Advanced Robotics Technology and Systems Laboratory) Pisa-Italy 56025 (phone: 303-555-5555; e-mail: lordkenneth.pinpin@sssup.it, fgamarra@sssup.it, cecilia.laschi@sssup.it and paolo.dario@sssup.it).

II. THEORETICAL BACKGROUND

A. *Adaptive Neuro-Fuzzy Inference System (ANFIS)*

The ANFIS [6] is reviewed here briefly. Adaptive Neuro-Fuzzy Inference Systems are Fuzzy Sugeno models put in the framework of adaptive systems, a fuzzy Sugeno type is composed by rules of the type:

Rule 1: if x1 is A1 and x2 is B1, then
f1 = a1x1+b1x1+c1
Rule 2: if x1 is A2 and x2 is B2, then
f2 = a2x1+b2x2+c2

Figure 1 illustrates the architecture of the network. In the first layer the degree of the membership of the input is computed using a Gaussian membership function:

$$\mu_{A_i} = \frac{1}{1+\left[\left(\frac{x-c_i}{a_i}\right)^2\right]^{b_i}} \tag{1}$$

Where a_i, b_i and c_i are the parameters of the Gaussian function. The second layer calculates the firing strength (or weight) w_i of the i_{th} rule,

$$w_i = \mu_i(x_1)\mu_i(x_2) \tag{2}$$

In the third layer the firing strengths are normalized with the sum of all rule's firing strengths:

$$\overline{w} = \frac{w_i}{w_1+w_2} \tag{3}$$

In the fourth layer the output is calculated as the product of the normalized firing rate and the parameters set:

$$\overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i) \tag{4}$$

Finally in the fifth layer is calculated the overall output as the addition of all incoming signals,

$$\sum_i \bar{w} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

Training the network consists of finding suitable parameters for layer 1 and layer 4. Gradient descent methods are typically used for the non-linear parameters of layer 1 while batch or recursive least squares are used for the linear parameters of layer 4 or even a combination of both. See [6] for details.

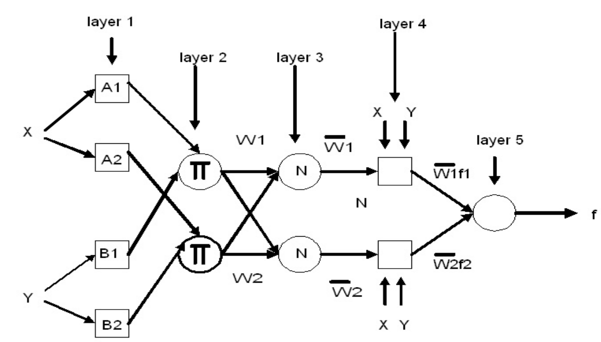

Fig. 1. ANFIS architecture.

B. *Visual Servoing*

The reaching task has been implemented using visual servoing. Visual serving is a technique that has been studied and used in robotics. In [7] we can find a survey about the methods and techniques used in visual servoing. The controller implemented in this paper is classified according to [7] as an Image Based Visual Servoing technique (IBVS) and eye-to-hand scheme.

The algorithm used for the visual servoing is the one based on Piepmeier [8]. Piepmeier used a dynamic Gauss-Newton method to minimize the errors in the image plane. The error for a static target is defined as the difference of the position in the image plane of the target y^* and the end-effector $y(\theta)$.

$$f(\theta) = y(\theta) - y^* \tag{6}$$

The dynamic Gauss-Newton method computes the joint angles iteratively. At each iteration k the angular position is computed as

$$\theta_{k+1} = \theta_k - (\hat{J}_k^T \hat{J}_k)^{-1} \hat{J}_k^T \left(f_k + \frac{\partial f_k}{\partial t} h_t \right). \tag{7}$$

The term h_t is a time increment and is defined as $h_t = t_k - t_{k-1}$; the term $\frac{\partial f_k h_t}{\partial t}$ predicts the change in the error function for the next iteration and $\hat{J}_k$ represents an approximation to the Jacobian in the k instant.

$$\hat{J}_k = \hat{J}_{k-1} + \left(\Delta f - \hat{J}_{k-1} h_\theta - \frac{\partial f_k}{\partial t} h_t \right) (\lambda + h_\theta^T P_{k-1} h_\theta)^{-1} h_\theta^T P_{k-1} \tag{8}$$

$$P_k = \frac{1}{\lambda} (P_{k-1} - P_{k-1} h_\theta (\lambda + h_\theta^T P_{k-1} h_\theta)^{-1} h_\theta^T P_{k-1}) \tag{9}$$

Where $0 < \lambda \leq 1$ is the forgetting factor, $h_\theta = \theta_k - \theta_{k-1}$, $\Delta f = f_k - f_{k-1}$. Equations (8) and (9) define the recursive update of $\hat{J}_k$.

III. General Architecture Description

Figure 2 shows the general architecture that is used to accomplish the reaching task. For a reaching task the first thing done was to develop a sensory-motor coordination map of the robot (forward model). This sensory-motor map coordination is constructed after a motor babbling phase. In the babbling phase the robot moves its joints randomly along the workspace and stores information that comes from its proprioceptive system as the readings from the encoders and information derived from the visual system.

The information obtained from the babbling phase is used to create a forward model of the robot. This forward model of the robot relates information of the angular joint positions, and end effector coordinates in the image plane. The forward model is constructed using an ANFIS neural network. The forward model is then used to initialize the image Jacobian of our visual servoing algorithm. As it has been described in the previous section the visual servoing algorithm is a closed loop algorithm based on the visual servoing method of [8] that controls the robot to reach a target.

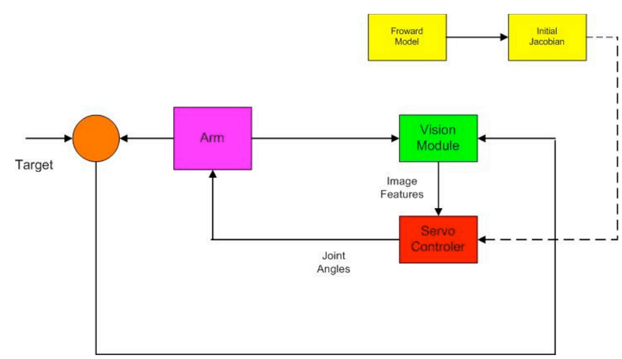

Fig. 2. General Architecture for a reaching task using a forward model.

IV. Motor Babbling

In this exploratory phase, the angular positions of the robot joints and the end-effector position in the visual system is stored. The Robotics Toolbox of Peter Corke for Matlab [9] was chosen as a simulation platform. The simulated puma 560 manipulator is used in this work. For the vision system the Epipolar Geometry Toolbox for Matlab [10] is used to simulate two fixed cameras.

A total of 720 samples was collected and divided into 600 samples for the training phase and 120 samples for the testing phase. There are two simulated cameras one for the right eye and the other one for the left eye. The internal parameters for the two cameras are the same, then the images in the two cameras are of 640*480 pixels in size and

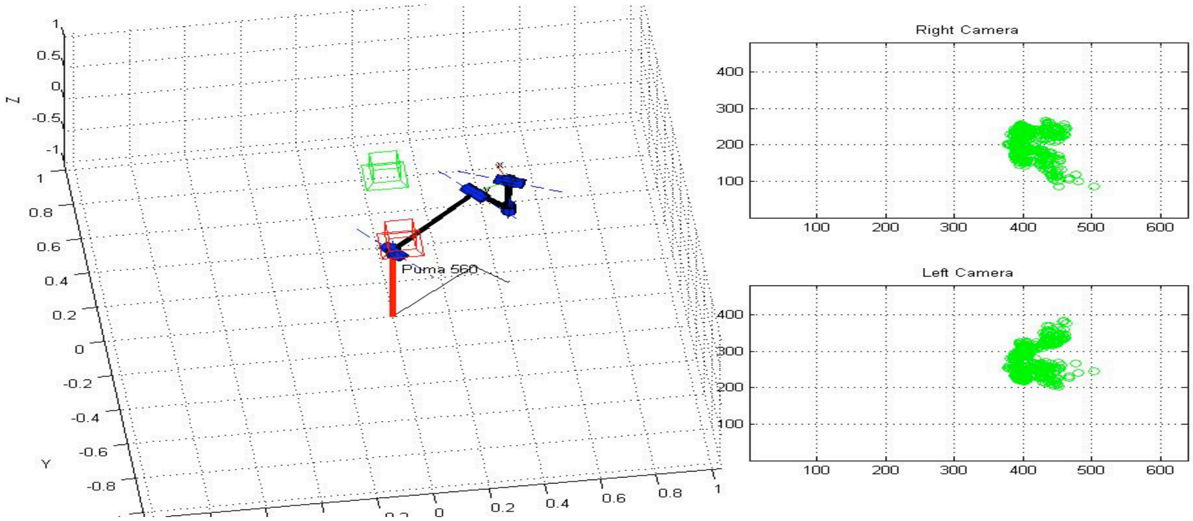

Fig. 3. The left subplot shows the position of the robot after the second babbling stage (joint 0 is initialized to 20 degrees). The top and bottom subplots show the imaged end-effector points at each babbling step.

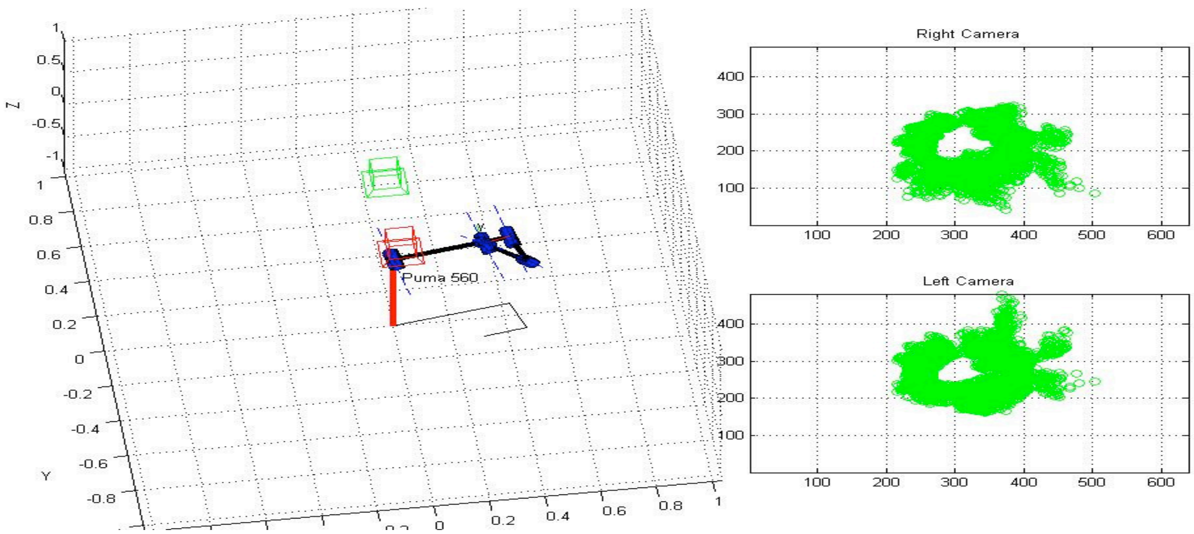

Fig. 4. The left subplot shows the position of the robot at the end of the babbling. The right top and right bottom subplots are the end effector accumulated positions of the end effector in the right and left camera.

the orthogonality factor of the CCD image axes is 0. The number of pixels per unit distance in image coordinates times the focal length is 50 for both cameras. Both cameras have a rotation of -90 degrees with respect to x axis and are located at [0, -0.2, 0.8] and [0, 0.2, 0.8] in the world coordinates.

To obtain a thorough sampling of the robot workspace the babbling was done in 18 stages. At the beginning of each stage the robot joint 0 is set to $20n$ degrees, where n is the stage number. Each stage is made of 40 babbling steps. Each joint is randomly perturbed with a maximum change of 3.82 degrees.

Figure 3 shows the robot and the resulting image points after the first babbling stage. Figure 4 shows the end of the 18 babbling stages and the resulting images of the end-effector points.

V. Forward Model Creation

The data collected from the babbling phase is used to create a forward model of the robot. Figure 5 shows how the forward model is constructed using the ANFIS toolbox of Matlab. The input data is a set that includes the end effector position in the image and joint angles of the manipulator. The input data is clusterized using the unsupervised clustering algorithm of the toolbox that uses the subclustering algorithm [11]. The unsupervised clustering algorithm gives the initial structure of the network (number of fuzzy rules and parameters for the initialization of the membership functions).

A total of four ANFIS neural networks have been constructed – one ANFIS for each image feature coordinate (u_L, v_L, u_R, v_R). Each neural network has 9 inputs (q_0, q_1, q_2, q_3, q_4, q_5, p_x, p_y, p_z). The first 6 inputs are the angular positions

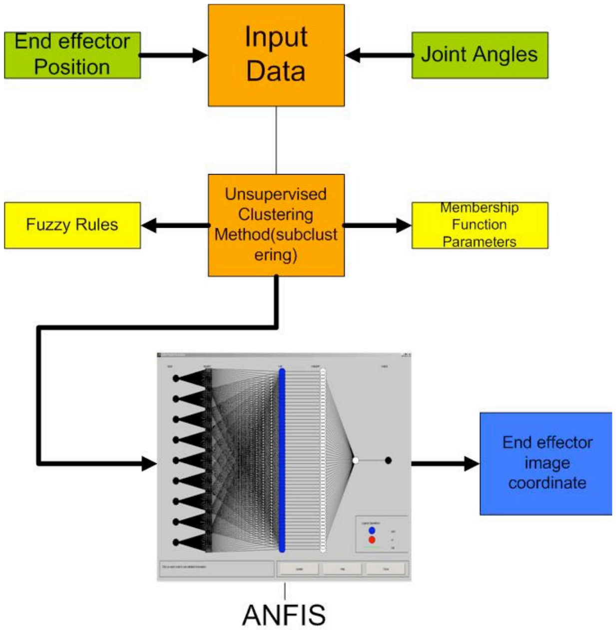

Fig. 5. Construction of the forward model using an ANFIS-based methodology.

of the joints of the manipulator and the other 3 inputs are the coordinates of an end-effector point (with respect to the end-effector local frame). The output of the network is a image coordinate (*u* or *v*) of the end-effector position (p_x, p_y, p_z) in one of the cameras (left or right). A total of five feature points have been tracked because it has been seen in the simulations that as the number of tracked points is increased the robustness of the algorithm grows.

Table 1 shows the number of rules that result with a subclustering radius of 0.55. This subclustering parameter tunes the number of fuzzy rules constructed. There is a tradeoff for chosing the value of this parameter because if there are more rules the computational burden increases. The training was done using a hybrid method that is a combination of back-propagation and recursive least square algorithms. The four ANFIS neural networks were trained just for 100 epochs. Table II shows the training error for the four neural networks, at the beginning of the training and after 100 epochs.

VI. Reaching Task

A. Initial Image Jacobian Estimation

The forward model encoded in the ANFIS networks is used in obtaining an estimate of the initial image Jacobian of the manipulator for a given joint position. To obtain the initial estimation of the Jacobian a virtual perturbation of the manipulator joints at the current position is done using the ANFIS networks. Each joint is individually perturbed and the resulting changes of the feature points are used to initialize the corresponding column of the image Jacobian. The changes in the image feature points are computed using the forward model instead of the cameras. That is, the joint angles and the coordinates of each of the five tracked points

TABLE I
ANFIS Initialization

Neural Network	Subclustering radius	Number of rules
ANFIS 1 (u_L)	0.55	30
ANFIS 2 (v_L)	0.55	37
ANFIS 3 (u_R)	0.55	30
ANFIS 4 (v_R)	0.55	33

TABLE II
ANFIS Neural Network Training Errors

Neural Network	Initial Training Error	Final training error (100 epochs)
ANFIS 1 (u_L)	8.19338	3.81745
ANFIS 2 (v_L)	6.00207	1.93243
ANFIS 3 (u_R)	8.19338	3.81745
ANFIS 4 (v_R)	5.54171	2.01348

are inserted as inputs to our forward model. The output of the forward model gives the position each of the end-effector points in the image planes of the “robot eyes”.

B. Visual Servoing with a Forward Model

The final objective of the robot is that it could reach a target in a specific position of its workspace. The initial estimate of the image Jacobian is used at the beginning of the visual servoing controller.

The manipulator starts practically in a position opposed to the target. The variation of the joint velocities of the manipulator has been clamped between -0.9 and 0.9 rad/sec. Uniform image noise between ±5 pixels is added to the image features. The sampling period for this simulation is fixed at 1/30 sec. The number of iterations for the control loop is equal to 600 iterations.

The robot has as initial coordinates in joint space [-3.0252 0.07757 -1.5126 0 0 0] radians and the desired position of the target in joint space coordinates is [0.93 0 0 0 0 0] radians. Figure 6 shows the initial position of the robot and the target. The other subplots of this figure display the representation of these points in the two cameras at the beginning of the reaching. Figure 7 shows the position of the robot when it has reached the target.

In order to test the validity of the use of the forward model to calculate an initial image Jacobian a comparison with random image Jacobians is made. The servo-controller is tested with 10 different random initial image Jacobians. Each element of random image Jacobian (a 20 x 6 matrix) is initialized with values from the range [-1 1]. Figure 8 shows the results obtained with these random image Jacobians (blue curves) and the one estimated using the forward model (red one). This simulation shows that the error obtained with the initial Jacobian derived from the ANFIS forward models has a better performance and is matched only by two random Jacobians. It possible to consider this estimation as an optimal estimation or a one that is close to the optimal estimation when compared with random estimations, which by virtue of being a random process, is no guarantee that a good image Jacobian can be found.

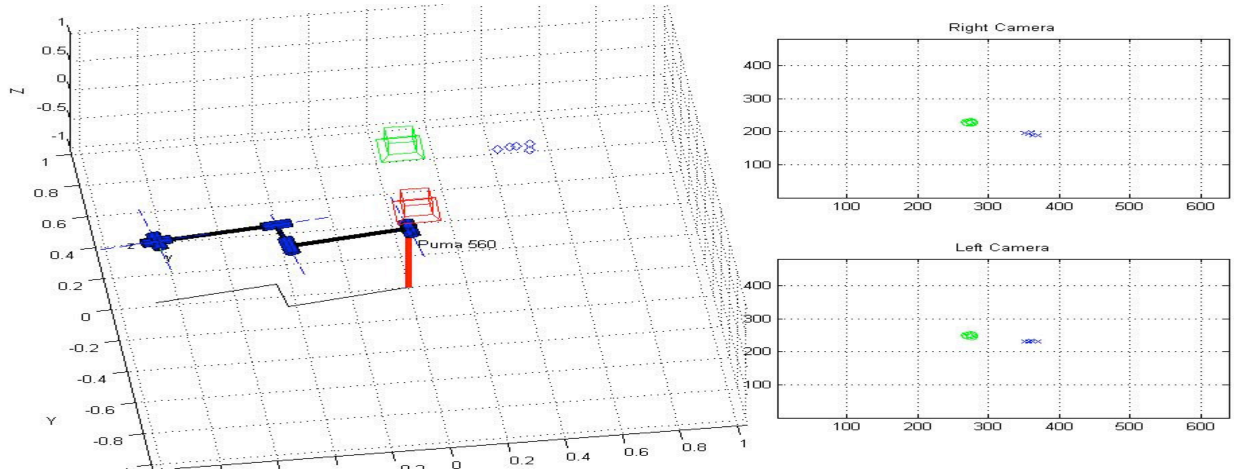

Fig. 6. The left subplot has the position of the robot at the beginning of the reaching. The target position is represented by the blue points in the left subplot (feature points of the end effector at the end of reaching). The right top and right bottom subplots show the end-effector position (green circles) and the target image coordinates (blue crosses).

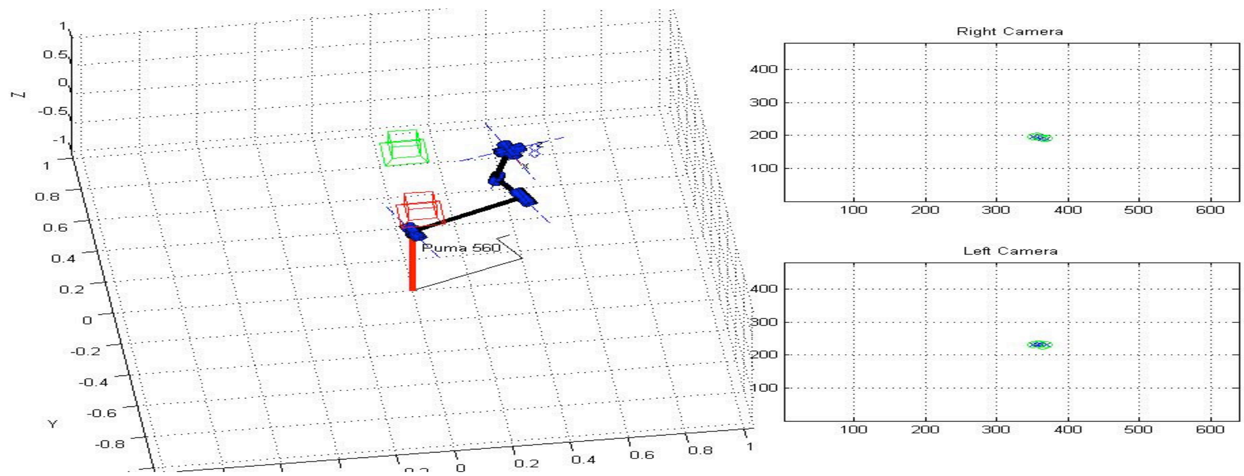

Fig. 7. The left subplot has the position of the robot at the end of the reaching. The target is represented by the blue points in the left subplot (partially obscured by the end effector). The right top and right bottom subplots show the end-effector position (green circles) and the target (blue crosses). Since the robot end-effector has reached the target the blue crosses are overlapping with green circles.

VII. Conclusion

This paper describes how through a babbling motor phase similar to a self exploratory "babbling body" process developed by infants a forward model is constructed. The forward model is constructed using ANFIS neural networks. The forward model created serves to initialize optimally the image Jacobian that is used in the image-based visual servoing controller. The results obtained in a reaching task using an estimated image Jacobian shows how an already robust visual servoing routine can be improved by a forward model.

This paper has demonstrated a new way in which a forward model can be used. We are currently in developing locally and globally better controllers using forward models.

VIII. Acknowledgment

This work is supported by the ROBOTCUB project (IST-2004-004370), funded by the European Commission through Unit E5 "Cognitive SYSTEMS".

References

[1] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," Trends in Cognitive Sciences, vol. 2, pp. 338-347, September 1998.

[2] M. Kawato, "Internal models for motor control and trajectory planning," Curr Opin Neurobiol, vol. 9, pp. 718-27, Dec 1999.

[3] R. P. N. Rao, A. P. Shon, and A. N. Meltzoff, " Bayesian model of imitation in infants and robots," in Imitation and Social Learning in Robots, Humans, and Animals: Behavioural, Social and Communicative Dimensions, K. Dautenhahn and C. Nehaniv Eds. Cambridge: Cambridge University Press, 2004.

[4] G. Sun and B. Scasellati, "A fast and efficient model for learning to reach," International Journal of Humanoid Robotics, vol. 2, pp. 24-32, 2005 2005.

[5] A. Dearden and Y. Demiris, "Learning Forward Models for Robots," in in Proceedings of IJCAI, Edinburgh, 2005.

[6] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," Systems, Man and Cybernetics, IEEE Transactions on, vol. 23, pp. 665-685, 1993.

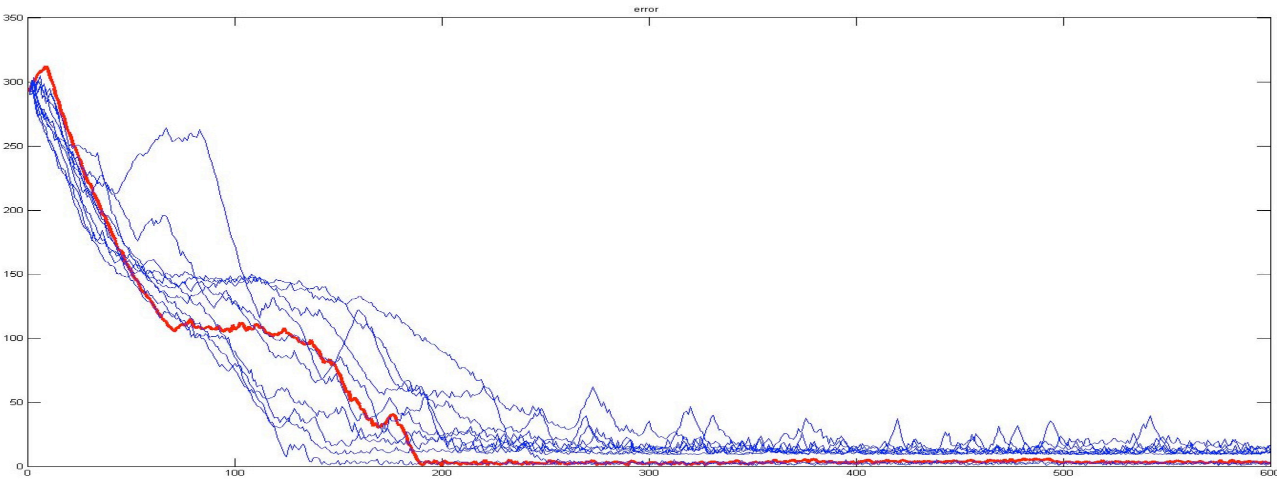

Fig. 8. Comparison of servoing error performance using random image Jacobians (blue) and the image Jacobian estimated from the ANFIS forward-model (red). Each curve represents the norm of the error vector in pixels at each time step.

[7] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," Robotics and Automation, IEEE Transactions on, vol. 12, pp. 651-670, 1996.

[8] J. A. Piepmeier, G. V. McMurray, and H. Lipkin, "A dynamic Jacobian estimation method for uncalibrated visual servoing," in Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on, 1999, pp. 944-949.

[9] P. I. Corke, "A robotics toolbox for MATLAB," Robotics & Automation Magazine, IEEE, vol. 3, pp. 24-32, 1996.

[10] G. L. Mariottini and D. Prattichizzo, "EGT for multiple view geometry and visual servoing: robotics vision with pinhole and panoramic cameras," Robotics & Automation Magazine, IEEE, vol. 12, pp. 26-39, 2005.

[11] R. Yager and D. Filev, "Generation of fuzzy rules by mountain clustering," Journal of Intelligent and Fuzzy Systems, vol. 2, pp. 209-219, 1994.