

Direct Visual Servo Control of a Robot to Track Trajectories in Supervision Tasks

Gabriel J. Garcia Carlos A. Jara Jorge Pomares Fernando Torres
Physics, Systems Engineering and Signal Theory Department
University of Alicante
P.O. Box 03080 Alicante, Spain
{gjjg, Carlos.Jara, jpomares, Fernando.Torres}@ua.es

Abstract—This paper describes the modelling, simulation and visual servo control of a robot-arm designed for supervision of complex task environments. This robot (mini-robot) has 2 degrees of freedom (hereafter d.o.f.) and it is attached at the end of another anthropomorphic robot with 7 d.o.f. (main robot). In order to achieve a suitable supervision, the mini-robot has to perform a given trajectory around the workspace where the task is developed. With the purpose of obtaining an accurate and fast supervision, this paper presents a novel tracking method carried out in the image space using direct visual servo control.

Keywords—direct visual servo, path tracking, visual servoing.

I. INTRODUCTION

Visual servoing systems permit to carry out point-to-point motion of a robot using visual information. The most known classification for this kind of control systems divides them in the position-based and the image-based visual servoing systems [1]. The first ones use 3D information of the real workspace to guide the robot and the second ones utilize 2D information provided by the image camera. This last approximation is the used for the purpose of this paper.

Image-based visual servoing systems are asymptotically stable only in a local region when the initial and desired positions of the robot are close [1]. However, in long distance movements, the errors generated in the extrinsic parameters computing [2] or the produced in the distance object estimation [3] can converge the robotic system to a local minimum [1]. Chesi and Vicino propose in [4] one of the first control law which can ensure global convergence and features in the field of view. Another possible solution for this problem is to achieve the final position following a desired sampled trajectory. These samples set, for each iteration of the control loop, a different reference which is employed as a desired position for the robotic system. In this way, the current and desired positions are very close, and therefore, the global system is stable due to the local stability of the image-based visual servoing systems. Several works have been detailed in the literature to deal with the problem of path planning in the image space [5][6]. This paper does not focus on path planning but on path tracking. In [7], a tracking method in the image space called time-independent path tracking based on virtual visual servoing is described. This method allows the robot to perform a time-independent tracking with a constant speed provided by the user. This path tracking system employs

classical image-based visual servoing, which assumes that the robot is a perfect positioning device. This type of control does not take into account the system dynamics, what is not suitable when the robot execute fast and/or accurate movements. As it is mentioned in [8], until the mid of 90s, few proposed visual-based controllers took into account the non-linear dynamic model of robot-arms. During the last 15 years, the research trend in this field continues being the same: controllers are designed with the assumption that the robot is a perfect positioning device without dynamics. By means of direct visual servo, the internal control loop of servo motors is removed, so the visual servo control has to stabilize the robot. One of the first research works about direct visual servo control was the developed in [9]. The proposed controller by Miyazaki and Masutani was based on the transpose Jacobian approach, method implemented for the first time by Takegaki and Arimoto [10]. However, in this approach, the visual system was modelled as a simple rotation transformation, without taking into account the robot translation. Kelly and Márquez [11] implemented a more accurate camera-robot system model than the proposed in [9]. This method solves the problems about the need to obtain the intrinsic parameters of the camera, but requires knowing accurately its orientation. This problem is solved in [8], where the controller proposed in [11] is improved to consider the generated uncertainty for the camera orientation allowing local asymptotic stability.

During the first decade of 21st century, it is studied in depth the adaptive control theory, which allows solving the errors of the dynamic parameters estimation. This way, Zergeroglu et al. design an adaptive controller which takes into account the estimation uncertainty of the camera-robot parameters [12]. In 2008, Wu and Li develop another adaptive controller which estimates the extrinsic and intrinsic parameters of the camera considering the dynamic model of the robot [13]. In this field, it is worth to mention the research works of Wang et al. [14], which are about to solve the visual control of a 3 d.o.f. robot using an adaptive algorithm that updates, in each iteration, the distance value between the camera and the object. This way, this method obtains a distance-independent interaction matrix.

Only a few works about direct visual servo include an image-based visual control approach, and neither of previous works permits to track a predefined trajectory in the image space (they only can be used for point-to-point positioning). This lack is the main purpose of this paper: to implement a

direct visual control of a robot for path tracking in order to supervise a specific task. Specifically, it describes the modelling, simulation and control of 2 d.o.f. robot (mini-robot) which is attached at the end of another anthropomorphic robot with 7 d.o.f (main robot). The mini-robot has to perform a given trajectory around the workspace where the task is developed in order to get a correct supervision. With the objective of obtaining an accurate tracking with a high velocity, a novel method for path tracking using direct visual servo is described in this paper.

This paper is organized as follows: Section II describes the physical characteristics of the mini-robot and its modelling. In Section III, the main aspects about direct visual servoing are explained. Section IV describes the novel method that allows performing path tracking in the image space using direct visual servo control. Section V shows several test cases of this novel method. Finally, some conclusions are discussed in Section VI.

II. MODELLING OF THE MINI-ROBOT

This paper describes the modelling, simulation and control of a robot with the aim of its early physical construction. This mini-robot is expected to build and to attach it at the end of a Mitsubishi PA-10 robot (main robot) as shown in Figure 1. Direct visual servo control will be implemented in the mini-robot in order to track a previously set trajectory for supervising a complex task performed by the main robot.

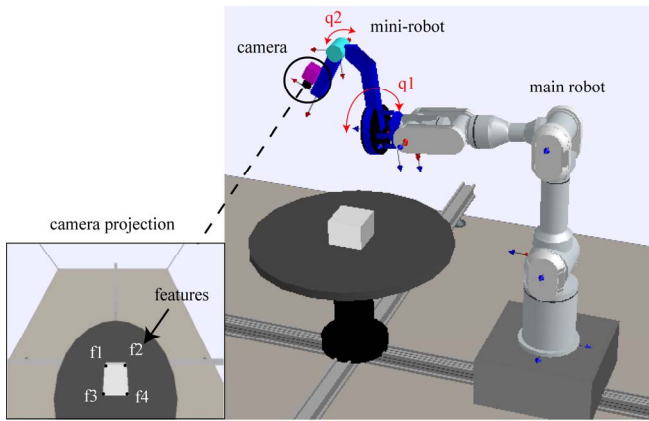


Figure 1. Virtual model of the robotic system

The robot has 2 rotational joints. The first link measures 15 cm long and its mass is about 0.5 kg. The second link measures 20 cm long and it weighs 1.2 kg including the mass of the end-camera (Figure 1). This paper is not focused in complex image processing issues. Therefore, the image trajectory is easily generated using the four corners of an object in the workspace which will be the extracted features for the visual servoing path tracking system $\mathbf{s}=(f1,f2,f3,f4)$.

III. DIRECT VISUAL SERVOING

Using direct visual servo control, the visual servoing controller is in charge of computing the suitable joint torques of the robot to reach the required position. The result is a robot control much faster and more accurate. Robot dynamics is concerned with the equations of motion for n -axis robot manipulators which reflect the relationship between the forces acting on a robot mechanism and the accelerations produced:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{G}(\mathbf{q})=\boldsymbol{\tau} \quad (1)$$

In this equation, $\mathbf{q} \in \mathbb{R}^{n \times 1}$ are the joint positions, $\dot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$ represent the joint velocities and $\ddot{\mathbf{q}} \in \mathbb{R}^{n \times 1}$ are the joint accelerations. Furthermore, \mathbf{M} is the symmetric joint-space inertia matrix, \mathbf{C} describes Coriolis and centripetal effects, \mathbf{G} is the gravity loading matrix and $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$ represents the joint torques.

In [15], it is described a direct visual servo control system based on the transpose Jacobian method, which was introduced in [10]. The proposed control law is the following:

$$\boldsymbol{\tau}=\mathbf{J}(\mathbf{q},\mathbf{s},\mathbf{Z})^T K_p \mathbf{e}-K_v \dot{\mathbf{q}}+\mathbf{G}(\mathbf{q}) \quad (2)$$

where $K_p \in \mathbb{R}^{2M \times 2M}$ and $K_v \in \mathbb{R}^{n \times n}$ are the proportional and derivative gains, respectively. These constants are symmetric positive-defined matrices which are estimated by the user. M is the number of image features extracted and n is the number of robot joints. Inside the first term of (2), $\mathbf{e}=(\mathbf{s}-\mathbf{s}_d)$ is the image error, where \mathbf{s} represents the visual features at the current state and \mathbf{s}_d denotes the desired visual features, i.e., the features observed by the camera at the desired robot pose. $\mathbf{J}(\mathbf{q},\mathbf{s},\mathbf{Z})^T$ is defined as the transpose of the Jacobian matrix and it is computed according the following equation [15]:

$$\mathbf{J}(\mathbf{q},\mathbf{s},\mathbf{Z})=\mathbf{L}_s(\mathbf{s},\mathbf{Z}) \cdot \mathbf{J}_g(\mathbf{q}) \in \mathbb{R}^{2M \times n} \quad (3)$$

where $\mathbf{L}_s(\mathbf{s},\mathbf{Z}) \in \mathbb{R}^{2M \times 6}$ is the image Jacobian for the desired features [16] and it depends on the current position of the robot joints since uses an eye-in-hand configuration. $\mathbf{J}_g(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ is the Jacobian matrix of the robot [17]:

$$\begin{bmatrix} v_c \\ \boldsymbol{\omega}_c \end{bmatrix}=\mathbf{J}_g(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (4)$$

The second term of (2) $-K_v \dot{\mathbf{q}}$ belongs to the derivate action and requires the joint velocity $\dot{\mathbf{q}}$. Finally, \mathbf{G} is the gravity loading matrix defined before in (1). Therefore, it is necessary to compute this matrix to obtain the joint torques to apply at the robot for minimizing the image error. This term of the dynamic model only depends of the current robot joint position \mathbf{q} .

The controller shown in (2) has local asymptotic stability when the current features \mathbf{s} , and the desired ones \mathbf{s}_d , are close.

In [15], this stability is demonstrated using the Lyapunov method. This controller is suitable for point-to-point tasks, but it is not able to do path tracking. Next section will be described a method that allows to perform path tracking in the image space using the dynamic controller shown in (2).

IV. DIRECT VISUAL SERVOING TO TRACK TRAJECTORIES

In [7], a visual servoing system for tracking trajectories based on indirect visual servoing is described. The idea of this tracking system is to change continuously the visual features used as references for a classical visual servo control. In this paper, this control strategy will be extended to apply it with direct visual servoing.

With the aim of doing path tracking by means of the control law expressed in (2), it is modified the error function with the way that references only depend on the current position of the camera $e = (s - s_d(\mathbf{q}))$. Thus, the next reference of the trajectory $s_d(\mathbf{q})$, is computed from the current camera position.

The robot has to follow a predefined path to perform the task, as explained above. Before the tracking process, the discrete trajectory of the features in the image to be tracked by the robot $T = \{^k s_d / k \in 1..N\}$ is obtained using our previous work [18], where $^k s_d$ are the set of M points or features observed by the camera at instant k , $^k s_d = \{^k f_{id} / i \in 1..M\}$. The set of visual features observed at the current camera position are represented by $^c s$. From this set of image features, it is necessary to find an image configuration which provides the robot with the desired velocity $|v_d|$ by iterating over the set T . For each image configuration $^k s$, the corresponding camera velocity is determined using (2):

$$\tau = \mathbf{J}(\mathbf{q}, \mathbf{s}, \mathbf{Z})^T K_p ({}^c \mathbf{s} - {}^k \mathbf{s}) - K_v \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) \quad (5)$$

Applying the forward dynamics from the previous torques and the relation described in (4), the camera velocity $|v|$ is obtained. This process continues until the velocity $|v|$ is obtained, which is greater than $|v_d|$. To obtain the set of features, $^j s$, which assures a constant velocity, the poses of the camera in 3D Cartesian space (from which $^k s$ and $^{k-1} s$ are observed) are computed using virtual visual servoing [19]. The camera pose from which $^j s$ is observed is calculated over the 3D line which links both camera poses by using the Levenberg-Marquardt non-linear optimization algorithm [20]. At this moment, the set of features $^j s$ will be the desired features s_d to be used by the controller presented in (2). This process is repeated until the given trajectory is completely tracked.

Using the Robotics Toolbox [21] and the Visual Servoing Toolbox [22], it has been developed the dynamic control law described in this section for path tracking. Figure 2 shows the *Simulink* diagram of this control law. From the robot joint position \mathbf{q} , the Jacobian matrix expressed in (4) is computed. The first term of the controller, $\mathbf{J}(\mathbf{q}, \mathbf{s}, \mathbf{Z})^T K_p e$, is calculated using this matrix. Afterwards, it is multiplied by the image Jacobian, this result is transposed and finally it is multiplied by K_p and by the image error. K_p is a symmetric matrix that has been adjusted in simulation to avoid oscillations and to perform a fast control. The second term of the controller (2), $-K_v \dot{\mathbf{q}}$, is computed multiplying the joint velocity that provides the forward dynamics by K_v (Figure 2). This matrix must be regulated to obtain a suitable response of the controller. The last term $\mathbf{G}(\mathbf{q})$ is computed from \mathbf{q} values using the function *gravload* of the Robotics Toolbox (Figure 2).

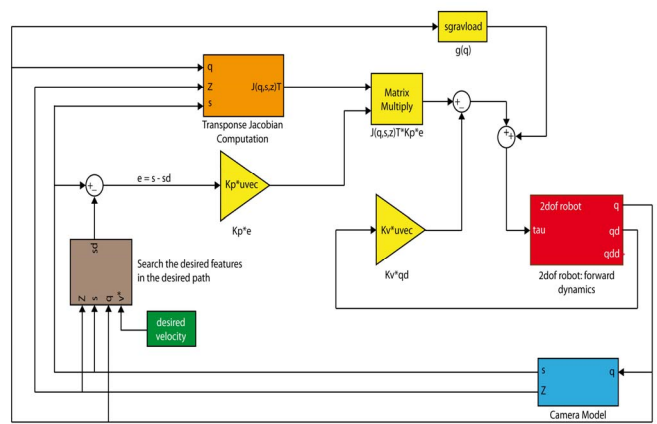


Figure 2. Simulink diagram of the novel path tracking method

V. RESULTS

In this section, two different test cases are described in order to verify the correct behaviour of the direct visual servoing controller proposed in this paper. In both cases, the given trajectory is successfully tracked.

A. Test 1

The first test describes the application of the method explained in Section IV for the path tracking of the trajectory shown in Figure 3. The module of the desired velocity is established to 2.7 mm/s. The trajectory starts in the initial position of the robot ($q_1 = 0$ rad, $q_2 = 0$ rad). This position sets the image features in $s_i = [219 \ 219, 294 \ 219, 294 \ 294, 219 \ 294]$ px. From this position, the first joint turns -0.4 rad and the second joint turns 0.15 rad. Thus, the final image position correspond with $s_f = [42 \ 419, 110 \ 390, 137 \ 459, 68 \ 488]$ px. In order to reach this final position, it is established to track the blue trajectory shown in Figure 3a. The chosen K_p and K_v constant values for this first test are the following:

$$K_p = 0.8 \cdot \mathbf{I}_{8 \times 8} \text{ Nm/px}^2 \quad (6)$$

$$K_v = \begin{pmatrix} 1.6 & 0 \\ 0 & 1.9 \end{pmatrix} \text{ Nms/rad.} \quad (7)$$

Figures 3a and 3b show the trajectory in image space of the features and the 3D trajectory performed by the robot, respectively. As it can be seen in these figures, the robotic system is able to track the trajectory in both spaces precisely. The system tracks the desired trajectory using the desired tracking velocity with a minimum delimited error in the image that does not affects the accurate tracking in the 3D Cartesian space.

Figure 4 shows the error evolution of the image features $e = (s - s_d)$ during the path tracking. At each iteration of the control loop, the suitable location of the desired image features is determined in order to track the trajectory with fixed velocity. Thus, the error in the image space is low and this guarantees the local stability of the system.

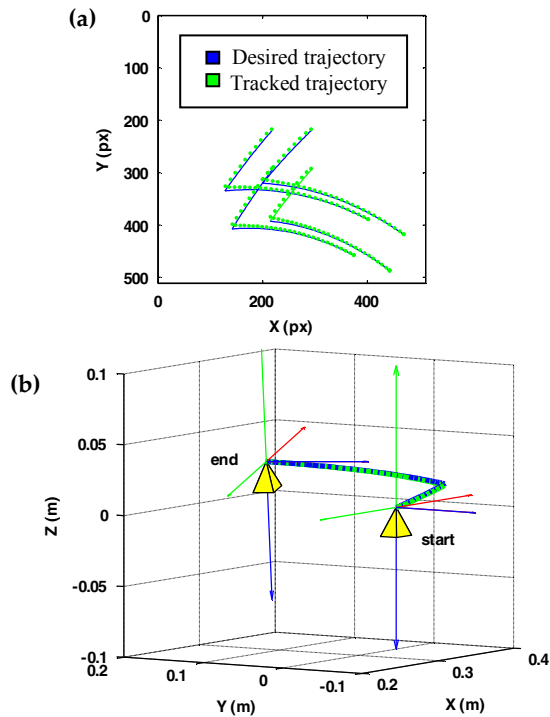


Figure 3. Test 1: Tracked trajectory a) in the image space. b) in the 3D space

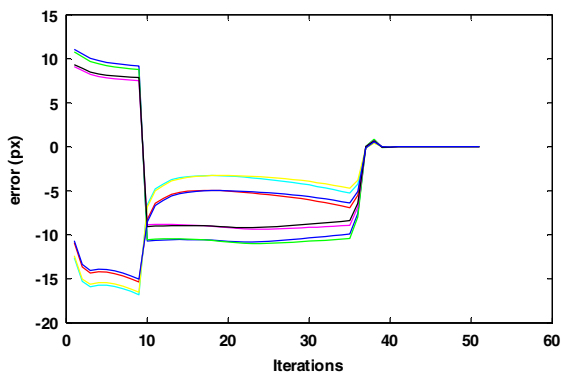


Figure 4. Test 1: Image error during the tracking.

Figure 5 represents the torque values sent to the two joints during the control path tracking of the defined trajectory. In the Figure 5a, it can be clearly seen negative torque values in a first part and positive torque values in a second part. In the temporal evolution of the second joint's torque also it can be seen this change in the trajectory when the slope varies in the evolution (see Figure 5b).

Figure 6 represents the lineal and angular velocity of the end of the robot during the path tracking. In the Figure 6b, it can be seen the temporal evolution of the velocity module. Both plots show how the velocity has been controlled during the tracking, maintaining it in a constant value (2.7 mm/s).

B. Test 2

In this test, the robot starts from its initial position ($\mathbf{q}=0$). As mentioned, this position sets the image features in $\mathbf{s}_i = [219 \ 219, 294 \ 219, 294 \ 294, 219 \ 294]$ px. From this position, the first joint turns 0.4 rad and the second joint turns 0.15 rad.

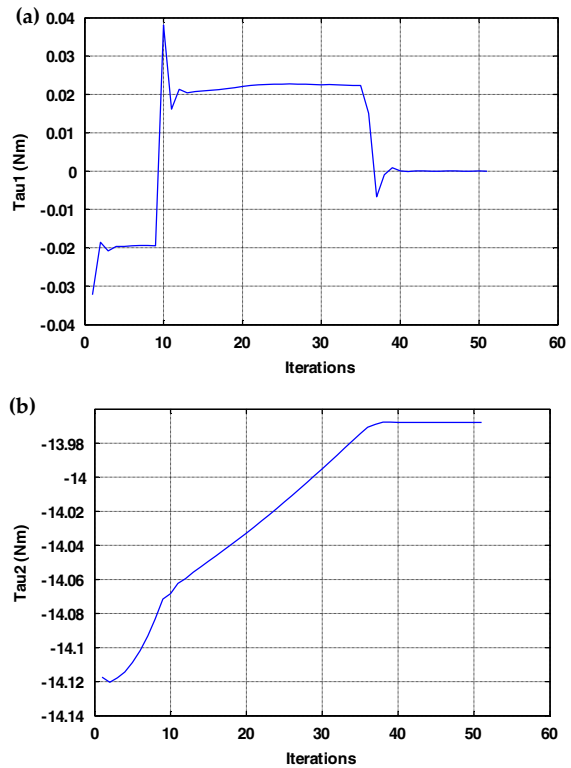


Figure 5. Test 1: Torque values sent by the controller for the path tracking to a) the first joint b) the second joint.

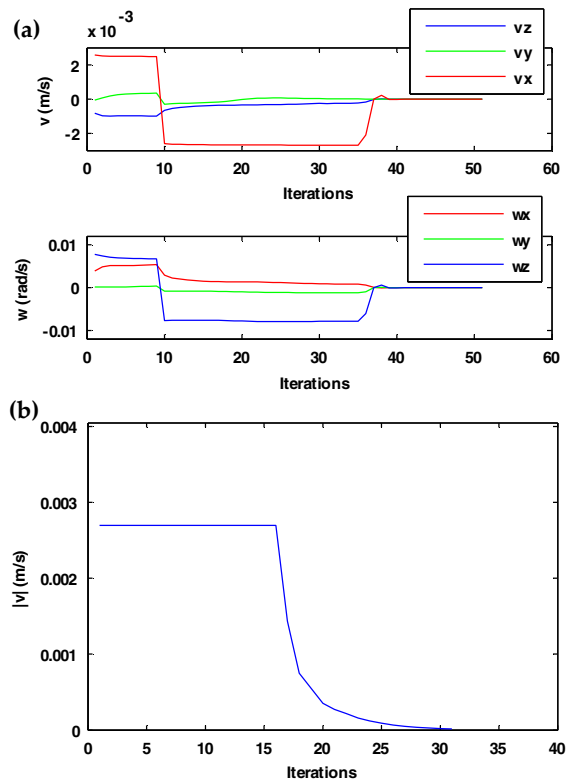


Figure 6. Test 1: a) lineal and angular velocity of the end of the robot during the tracking b) temporal evolution of velocity module during the tracking

Thus, the final image position correspond with $s_f = [402 \ 390, 470 \ 419, 444 \ 488, 375 \ 459]$ px. Between the initial and final point, the robot must track the predefined trajectory in the image space shown in Figure 7a.

The chosen K_p and K_v constant values for this second test are the following:

$$K_p = 1.9 \cdot \mathbf{I}_{8 \times 8} \text{ Nm/px}^2. \quad (8)$$

$$K_v = \begin{pmatrix} 4.6 & 0 \\ 0 & 3.6 \end{pmatrix} \text{ Nms/rad.} \quad (9)$$

This second test tries to evaluate the system behavior when the tracking velocity is higher. In the previous test, the tracking velocity was 2.7 mm/s. This velocity value allows reaching all the references of the desired trajectory. For this test, the module of the lineal velocity is set in 20 mm/s. Since the distance that the end of the robot must cover is about 160 mm (see Figure 7b), the task will be finished in approximately 8 seconds. This time is more suitable for the robot movement than the 55 seconds employed with a velocity of 2.7 mm/s (Test 1).

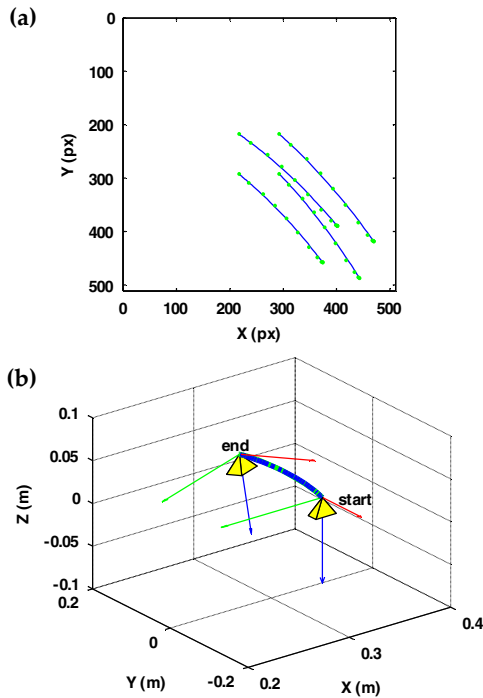


Figure 7. Test 2: Tracked trajectory. a) in the image space. b) in the 3D space

For these conditions, the resulting image trajectory is shown in Figure 7a. On the one hand, the image and Cartesian 3D trajectories (see Figure 7b) are adjusted accurately to the desired ones. On the other hand, because a higher velocity than the previous test, the error evolution in the image space presents a higher value in the first iterations (see Figure 8). The error in the image depicted in Figure 8 is computed at each iteration as the difference between the position of the current features and the position of the desired features in the image calculated with the method proposed here. Thus, it must be remarked that the higher the desired tracking velocity, the

greater the distance between current and desired features. And thus, the error represented in Figure 8 increases. Therefore, this parameter cannot be used as a performance quality measurement.

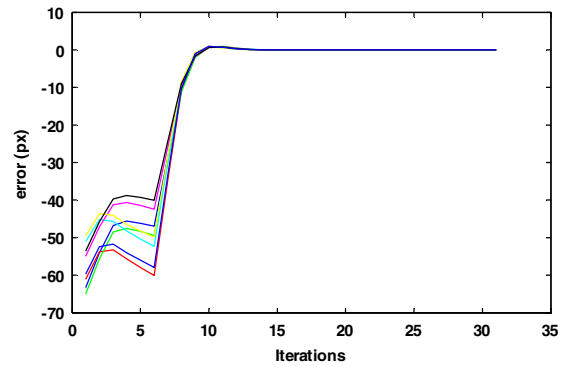


Figure 8. Test 2: Image error during the tracking.

In the Figure 9, it can be seen the joint torque values sent to the first and second joints. Figure 10 shows the control of the desired velocity, which was established in 20 mm/s. The lower number of iterations needed allow achieving the final position in a precise way and maintaining constant the lineal velocity. Thus, the path tracking is completely performed in less than 10 seconds (each iteration last 1 second, see Figure 10).

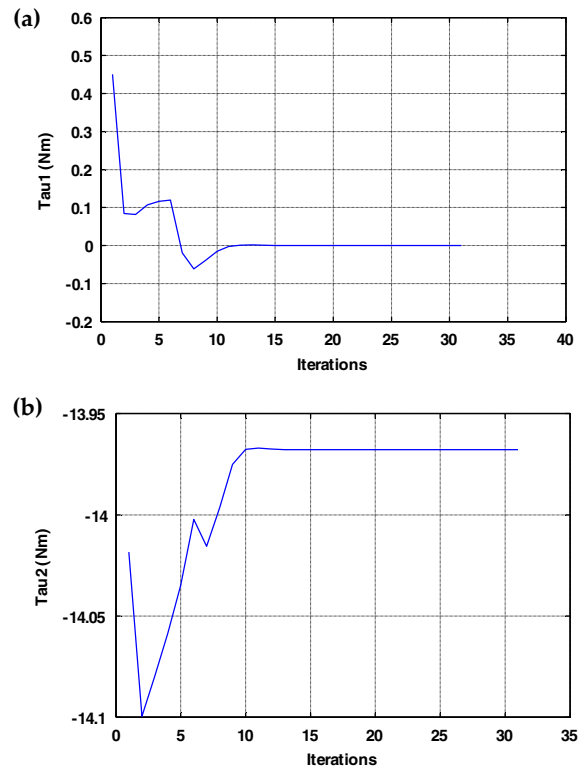


Figure 9. Test 2: Torque values sent by the controller for the path tracking to a) the first joint b) the second joint.

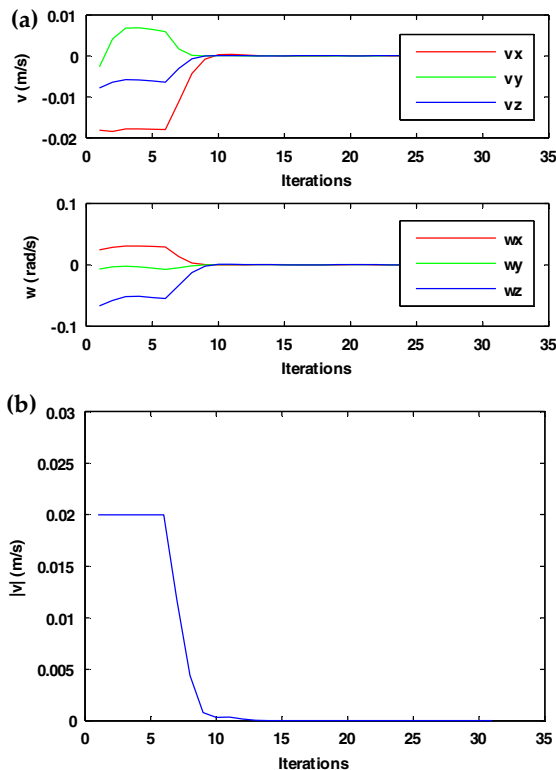


Figure 10. Test2: a) lineal and angular velocity of the end of the robot during the tracking b) temporal evolution of velocity module during the tracking

VI. CONCLUSIONS

This paper describes the modelling and control of a mini-robot which will be employed in supervision tasks. In order to perform the supervision, the robot must be able to track a given trajectory using image features extracted from the workspace. This paper presents, on the one hand, the modelling on the robot, and, on the other hand, the robot control. To implement this last issue, a new direct visual servoing system based on the transpose Jacobian matrix is presented.

In order to guarantee the local stability of the image-based visual servoing system, it is necessary that the current and the desired image features are close. However, this last aspect cannot be assured when the robot must perform large displacements. In order to avoid this problem, a novel direct visual servoing system to track trajectories is defined in this paper. The simulation results of this new method show that the current and the desired image features are always close during the path tracking. Once the path tracking method has been validated in several simulation tests, the next work will focus on the implementation of the controller in a first prototype of the mini-robot described in Section II. The results are expected to be confirmed in the control of this real robot because all the dynamic parameters of the robot are precisely known.

ACKNOWLEDGMENT

This work is supported by the Spanish Ministry of Education and Science (MEC) through the research project DPI2008-02647.

- [1] Chaumette, F.; Hutchinson, S. A., "Visual servo control. I. Basic approaches". *Robotics & Automation Magazine, IEEE*, 13(4) pp. 82-90, 2006
- [2] Kim, D. H.; Ryu, Y. G.; Chung, M. J., "Stability analysis of image-based visual servoing regarding with camera calibration errors", In *Proceedings of the 13th International Conference on Advanced Robotics*, Jeju, Korea, August 2007; pp. 176-181.
- [3] Malis, E.; Rives, P., "Robustness of image-based visual servoing with respect to depth distribution errors", In *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003; pp. 1056-1061.
- [4] Chesi, G.; Vicino, A., "Visual servoing for large camera displacements". *IEEE Transactions on Robotics*, 20(4) pp. 724-735, 2004.
- [5] Chesi, G., "Visual servoing path-planning via homogeneous forms and LMI optimizations". *IEEE Transactions on Robotics*, 25(2) pp. 281-291, 2009.
- [6] G. Chesi, K. Hashimoto (Eds). *Visual Servoing via Advanced Numerical Methods*. Springer, 2010, pp. 422.
- [7] Garcia, G. J.; Pomares, J.; Torres, F., "Automatic robotic tasks in unstructured environments using an image path tracker". *Control Engineering Practice*, 17(5) pp. 597-608, May 2009.
- [8] Kelly, R., "Robust asymptotically stable visual servoing of planar robots". *IEEE Transactions on Robotics and Automation*, 12(5) pp. 759-766, 1996.
- [9] Miyazaki, F.; Masutani, Y., "Robustness of sensory feedback control based on imperfect Jacobian", In *Proceedings of the fifth international symposium on Robotics research*, 1990; pp. 201-208.
- [10] Takegaki, M.; Arimoto, S., "A new feedback method for dynamic control of manipulators". *Journal of Dynamic Systems, Measurement, and Control*, 103(2) pp. 119-125, 1981.
- [11] Kelly, R.; Marquez, A., "Fixed-eye direct visual feedback control of planar robots". *Journal of Systems Engineering*, 5(4) pp. 239-248, 1995.
- [12] Zergeroğlu, E.; Dawson, D. M.; de Querioz, M. S.; Behal, A., "Vision-based nonlinear tracking controllers with uncertain robot-camera parameters". *Mechatronics, IEEE/ASME Transactions on*, 6(3) pp. 322-337, 2001.
- [13] Wu, B.; Li, H. G., "Uncalibrated visual servoing of robots with new image Jacobian estimation method". *Xitong Fangzhen Xuebao / Journal of System Simulation*, 20(14) pp. 3767-3771, 2008.
- [14] Wang, H. S.; Liu, Y. H.; Zhou, D. X., "Adaptive visual servoing using point and line features with an uncalibrated eye-in-hand camera". *IEEE Transactions on Robotics*, 24(4) pp. 843-857, Aug 2008.
- [15] Kelly, R.; Favela, J.; Ibarra, J. M.; Bassi, D., "Asymptotically stable visual servoing of manipulators via neural networks". *Journal of Robotic Systems*, 17(12) pp. 659-669, 2000.
- [16] Hashimoto, H.; Kubota, T.; Sato, M.; Harashina, F., "Visual control of robotic manipulator based on neural networks", *IEEE Transactions on Industrial Electronics*, 39(6) pp. 490-496, 1992.
- [17] Sciavicco, L.; Siciliano, B.; Sciavicco, B., "Modelling and control of robot manipulators". Springer-Verlag: London, United Kingdom, 2000; pp. 402.
- [18] Gil, P.; Pomares, J.; Puente, S.T.; Candelas, F.A.; García, G.J.; Corrales, J.A.; Torres, F. "A cooperative robotic system based on multiple sensors to construct metallic structures". *The International Journal of Advanced Manufacturing Technology*, 45(5), pp. 616-630, 2009
- [19] Marchand, É.; Chaumette, F., "A New Formulation for Non-Linear Camera Calibration Using Virtual Visual Servoing"; *Research Report no.1366*; IRISA: Rennes, 2001.
- [20] Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares". *Quarterly of Applied Mathematics*, 2(2) pp. 164-168, 1944.
- [21] Corke, P. I., "Visual control of robots : High-performance visual servoing". Research Studies Press: Taunton, England, 1996.
- [22] Cervera, E., "Visual servoing Toolbox for Matlab/Simulink", <http://vstoolbox.sourceforge.net>, 2002.