Dynamic Visual Servoing for Ping-Pong Game of a 3DOF PKM

Alberto Traslosheros, José María Sebastián, Lizardo Pari, Flavio Roberti, Ricardo Carelli

***Abstract*—This article describes the new visual servo control and strategies that are utilized to carry out dynamic tasks by the system Robotenis. This platform is a parallel robot that is equipped with an acquisition and processing system of visual information. Its main feature is that it has a completely open architecture control, planned in order to design, implement, test and compare control strategies and algorithms (visual and actuated joint controllers). Following sections describe a new visual control strategy specially designed to track dynamic objects in 3D space. Contrasting the strategies shown in previous works, where the end effector of the robot keeps a constant distance from the tracked object, in this work the controller is specially designed in order to allow changes the tracking reference. Changes in the tracking reference can be utilized to grip an object that is under movement or as in this case, Ping-Pong playing. Lyapunov stability is taken into account in the controller design.**

I. Introduction

Vision systems are more frequently incorporated in robotics applications. An important advantage of visual systems is that allow obtaining information from workspace and objects in tasks where contact measurements needs to be avoided. This visual information must be processed in order to extract useful information, in robotics areas is particularly interesting the use of position and orientation of objects in the robot task. Integration of visual systems and robots in dynamic tasks presents several unresolved issues that have been research object in important research centers [1][2]. Those issues include the design of control strategies of robotic high speed visual tracking tasks, such as in the Tokyo University ([3], [4] and [5]) where fast tracking (up to $2\ m/s$) strategies in visual servoing have been developed.

In order to study and implement different strategies of visual control, the Computer Vision Group at the Polytechnic University of Madrid decided to design and implement the Robotenis system. The platform (shown in Fig. 1) is a parallel robot of three degrees of freedom inspired in the delta robot, with an open control structure and equipped with a system of computer vision for image acquiring and analysis. The system has been built in order to interact with mobile objects that are in dynamic environments to carry out high-speed dynamic tasks. This article presents a new visual servo control algorithm for dynamic tracking.

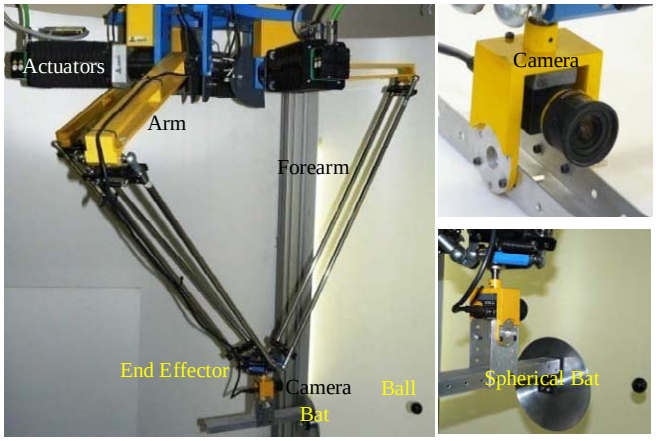

Fig. 1 Robotenis system.

Roughly speaking, a parallel robot consists of two platforms linked by more than one closed kinematic chain. This kinematic structure has several advantages over serial robots: high structural stiffness, high accuracy for end effector positioning, high load operation, high speed of the end effector and high acceleration, low engine and system inertia. On the other hand, the main disadvantage is the reduced range of its workspace.

Mechanical system of the Robotenis system is inspired by the DELTA robot [6], its kinematic model, the Jacobian matrix and its optimal design have been presented in previous work [7].The kinematic structure of the robot has been optimized and its dynamics has been studied in ADAMS. Therefore the design method solves two problems: The determination of the dimensions of the robot and the selection of the actuators. Dynamic analysis and joint control have been presented in Angel [8] and [9]. The control system consists of two control loops: the first one controls the actuated joints and is executed every $0.5\ ms$; the dynamic model is incorporated in the feedback control algorithm and additionally joint controller is based on a PD algorithm. The dynamics model is based on Lagrange multipliers and it is incorporated in the development of control strategies in the joint control loop. Second control loop is external to the first one, is executed every $8.4\ ms$ and is based on visual information. Second loop is the study object of present work.

The paper is organized as follows: After this introduction, in the second section the system Robotenis is briefly described, third section describes the proposed visual control algorithm and implementation issues, fourth section

This work was supported in part by the Universidad Politécnica de Madrid and International Cooperation Spanish Agency (AECI), A/026481/09.

A. Traslosheros, J. M. Sebastián and L. Pari are with the Technical University of Madrid, DISAM, José Gutierrez Abascal 2, CO 28003 Spain. Phone (+34) 913 363 061, email: atraslosheros@etsii.upm.es, jsebas@etsii.upm.es and lpari@etsii.upm.es.

F. Roberti and R. Carelli are with the National University of San Juan, Av. San Martín Oeste 1109, CO 5400 Argentina. Phone (+54) 0264 4 213 303, email: froberti@inaut.unsj.edu.ar and rcarelli@inaut.unsj.edu.ar. Department of Automatics, National University of San Juan, Argentina.

describes some of the results concerning to the measurement of the position of the ball and practical considerations to be taken into account when the controller is implemented, finally, fifth section gives some conclusions and results.

II. System Description

This section describes the Robotenis system, its functional characteristics, elements and its test environment. More information can be found at [10].

A. Test Environment

In this paper, main objective is the design of a visual controller that allows playing Ping-Pong. In order to do these task, and as initial approach, environment is strongly simplified, in future works environment complication is considered. To test the algorithm, a ping pong ball is hanging on a fixed structure (Fig. (2)). Ball is able to move at a speed close to 1 m/s. Environment simplification is done with a black ball on a white background.

B. Visual System

Robotenis system has a camera on the end of the robot, Fig (2). Camera location responses two main purposes: When the object is far from the end effector, the camera has a wide field of view. Second objective is that when the object and end effector are near each other noise has less effect over measurements and object trajectory can be accurately estimated. Those characteristics are important to our Ping-Pong application. Visual system basically consists of a light weight camera and a frame grabber. The camera SONY HCHR50 allows 240×640 pixels images at a sample time of 8.4 ms (binning mode). Progressive scan and integration time is 1 ms. Frame grabber is a Matrox Meteor 2-MC/4 which allows double buffer mode acquisition, essential to reach our visual sample time.

C. Image Processing

Once that image is acquired, image segmentation is done under knowledge of the ball on white background. Image features are taken from two image analysis: The first is calculated from the centroid of the ball projection and the second is the diameter of the projected ball that in measured from the image using sub-pixel accuracy. Spatial measurements are firstly given in the camera coordinates reference and in a second step are transformed to end effector coordinates. Camera is calibrated ([11]) and X Y coordinates are given from the position of the ball in the image plane, Z coordinate is calculated from the diameter of the ball. Control algorithms imply knowledge of the position and speed of the ball, which is estimated using a Kalman filter [12].

D. Actuators and Joint Control System

Join control system mainly consists of a DSPACE 1103 card where all robot algorithms are executed (implemented in ANSI C). Algorithms that are executed are the trajectory planning, kinematics and dynamics computation, Dynamic and PD controller, Kalman filter (ball position and speed). The motion system is composed of three 4 kw AC brushless servomotors, Ac drivers (Unidrive) and planetary gearboxes.

E. System Considerations

Robotenis system, its design, architecture and present application determine the proposed visual controller. In the design of visual controller it is necessary to take into account special characteristics of the system, as can be: noise in data that are acquired by the visual system, visual data sample time, visual data delay, actuators saturations, etc. Some of these characteristics can be a real challenge if are not properly taken into account. For example, variations in image measurements of 0.25 pixels can produce an error in the estimation of the velocity of the ball near to 1 m/s (when the ball is located to 600 mm far from the camera in the z axis).

In the Robotenis system sample delay in the visual loop is estimated in 2 samples T ($2T = 16.7\ mm$), visual delay is motivated by processes as can be integration of image in the camera, data transmission and image processing. Saturations of the actuators are taken into account in the design of a trajectory planner; maximum Jerk, maximum acceleration and maximum speed are taken into consideration [13].

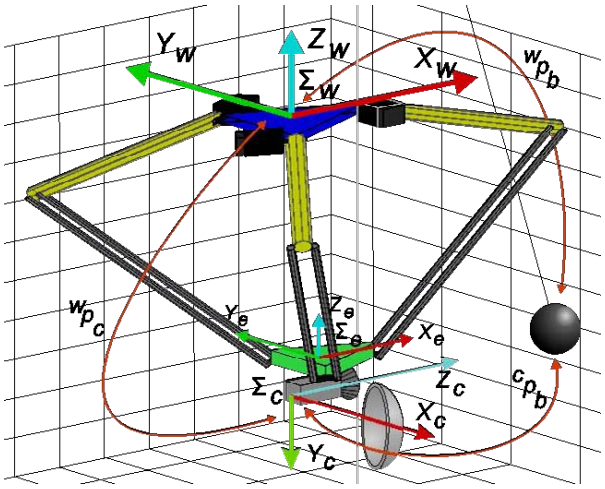

Fig. 2 Reference frames in Robotenis system.

III. Visual Servo Control

In order to design the visual controller this section will start by defining coordinated reference frames that are shown in Fig. 2, where Σ_w , Σ_e and Σ_c are coordinate frames in the world, in the end effector of the robot and in the camera respectively. Position of the end effector of the robot is defined in the word coordinate frame as ${}^{w}p_e$ (which is known through the robot direct kinematics), ${}^{c}p_b$ is the position of the ball in the camera coordinate frame and ${}^{w}p_b$ is the position of the ball in the world coordinate system. In this paper matrix transformations T and rotations R as can be: ${}^{w}R_e$, ${}^{w}R_c$ and ${}^{e}R_c$ and ${}^{e}T_c$ are supposed known through system modeling and calibrations of the system.

Although a wide range of controllers proposals are well established [14][15], in this article presents a visual tracking controller that is based in a camera in hand structure that is called: dynamic position-based look-and-move [16]. In order to solve our problem PBVS scheme permits that system can utilizes only one camera since the geometry of the object is know. As can be seen in Fig. 3 error can be obtained as the difference between the desired position of the ball (${}^{c}p_b^*(k)$) and the position that is obtained through the vision system (${}^{c}p_b(k)$), both in the camera reference frame and k is the sample time. In previous works desired position (${}^{c}p_b^*(k)$) is fixed [17], in this paper desired position is variable, and signal control and stability are obtained under this new assumption. As a result the visual controller sets the desired velocity of the end effector in order to decrease the error. Joint control signal is obtained from the visual control signal through the trajectory planner and the robot Jacobian.

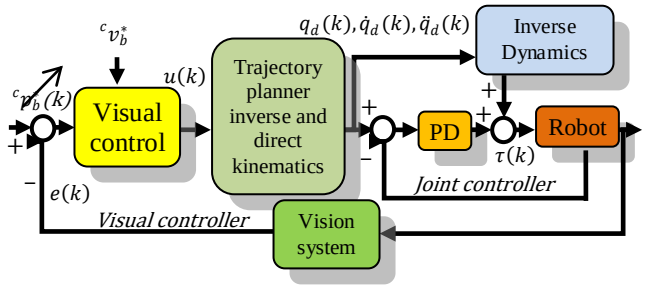

Fig. 3 Basic scheme of the controllers of the system.

A. Visual Controller Design

From Fig. 3 error can be defined as:

$$e(k) = {}^{c}p_b^*(k) - {}^{c}p_b(k) \tag{1}$$

Where ${}^{c}p_b^*(k)$ is the desired position and ${}^{c}p_b(k)$ is the measured position (measured by the visual system), both in the camera coordinate frame. Eq. (1) can be expressed.

$$e(k) = {}^{c}p_b^*(k) - {}^{c}R_w\left({}^{w}p_b(k) - {}^{w}p_c(k)\right) \tag{2}$$

Differentiating eq. (2), we obtain:

$$\dot{e}(k) = {}^{c}v_b^*(k) - {}^{c}R_w\left({}^{w}v_b(k) - {}^{w}v_c(k)\right) \tag{3}$$

From equations (2) and (3) it can observed that ${}^{c}p_b^*(k)$ is not a fixed position, thus its derivate is not zero and ${}^{c}v_b^*(k)$ will be the desired speed of approach. The desired speed of approach speed it is used to hit the ball with a desired velocity ${}^{c}v_b^*(k)$ and can be calculated from.

$${}^{c}v_b^*(k) = \left({}^{c}p_b^*(k) - {}^{c}p_b^*(k-1)\right)/T \tag{4}$$

In order to try to ensure an exponential decoupled decrease of the error:

$$\dot{e}(k) = -\lambda e(k) \tag{5}$$

Where $\lambda > 0$ and is a constant. On the other hand substituting (1) and (3) in (5) we obtain:

$$\begin{aligned} {}^{c}v_b^*(k) - {}^{c}R_w\left(\left({}^{w}v_b(k) - {}^{w}v_c(k)\right)\right. \\ = -\lambda\left({}^{c}p_b^*(k) - {}^{c}p_b(k)\right) \end{aligned} \tag{6}$$

Clearing the controller speed, we can obtain:

$$\begin{aligned} {}^{w}v_c(k) = {}^{w}v_b(k) \\ - {}^{c}R_w^T\left(\lambda\left({}^{c}p_b^*(k) - {}^{c}p_b(k)\right) + {}^{c}v_b^*(k)\right) \end{aligned} \tag{7}$$

Distance between end effector of the robot and camera is fixed thus velocity of the camera is equal that the end effector velocity. On the other hand measurements of the visual system are highly noisy and visual controller utilizes estimate data (${}^{c}\hat{p}_b(k)$, ${}^{w}\hat{v}_b(k)$) and the controller output is:

$$\begin{aligned} u(k) = {}^{w}\hat{v}_c(k) = {}^{w}\hat{v}_b(k) \\ - {}^{c}R_w^T\left(\lambda\left({}^{c}p_b^*(k) - {}^{c}\hat{p}_b(k)\right) + {}^{c}v_b^*(k)\right) \end{aligned} \tag{8}$$

B. Parameter λ

Once the controller in eq. (8) is obtained, the next step is the calculation of the parameter λ. For a sample period small enough it is possible to approximate.

$${}^{w}p_b(k+n) \cong {}^{w}p_b(k) + nT\,{}^{w}v_b(k) \tag{9}$$

In similar form:

$${}^{w}p_c(k+n) \cong {}^{w}p_c(k) + nT\,{}^{w}v_c(k) \tag{10}$$

and

$${}^{c}p_b^*(k+n) \cong {}^{c}p_b^*(k) + nT\,{}^{c}v_b^*(k) \tag{11}$$

For an error zero in the eq. (2) in the instant $k+n$, we have:

$$0 = {}^{c}p_b^*(k+n) - {}^{c}R_w\left({}^{w}p_b^*(k+n) - {}^{w}p_c(k+n)\right) \tag{12}$$

Substituting (9), (10) and (11) in (12):

$$\begin{aligned} 0 = {}^{c}p_b^*(k) + nT\,{}^{c}v_b^*(k) \\ - {}^{c}R_w\left({}^{w}p_b(k) - {}^{w}p_c(k) + nT\left({}^{w}v_b(k) + {}^{w}v_c(k)\right)\right) \end{aligned} \tag{13}$$

From above expression it is possible to attain:

$$\begin{aligned} nT\,{}^{w}v_c(k) = nT\,{}^{w}v_b(k) - {}^{c}R_w^T\left({}^{c}p_b^*(k) + nT\,{}^{c}v_b^*(k)\right) \\ + {}^{w}p_b(k) - {}^{w}p_c(k) \end{aligned} \tag{14}$$

Taking into account that ${}^{c}p_b(k)$ and ${}^{w}v_b(k)$ have to be estimated, and that ${}^{w}\hat{p}_b(k) - {}^{w}p_c(k)$ can be expressed as ${}^{c}R_w^T\left({}^{c}\hat{p}_b(k)\right)$ and substituting in (14) we have:

$$\begin{aligned} {}^{w}\hat{v}_c(k) = {}^{w}\hat{v}_b(k) \\ - {}^{c}R_w^T\left(\frac{1}{nT}\left({}^{c}p_b^*(k) - {}^{c}\hat{p}_b(k)\right) + {}^{c}v_b^*(k)\right) \end{aligned} \tag{15}$$

From eq. (8) and comparing to eq. (15) we can see that a good value for λ can be $\lambda = 1/(nT)$ for a "n" small enugh.

C. Visual Controller Stability

In this section stability of the visual controller is probed by means of Lyapunov theory. It is possible to probe that under ideal conditions the error converges to zero but if conditions are not ideal then it can be proved that error is finally bounded. In order to analyze the controller stability we start regarding equations (3) and (7) in order to attain the following closed loop expression:

$$\dot{e}(k) = -\lambda e(k) \tag{16}$$

We choose as Lyapunov function:

$$V = \frac{1}{2} e^T(k) e(k) \tag{17}$$

And from (17) and (16):

$$\dot{V} = e^T(k)\dot{e}(k) = -e^T(k)\lambda e(k) \tag{18}$$

We can see that eq. (18) is always negative ($\lambda > 0$) but we know that ${}^w v_b \equiv u$ is not completely true and this implies that $\dot{e}(k)$ is not exactly equal to $\lambda e(k)$ and eq. (18) is not completely fulfilled. For this reason and it is important to consider an error ρ.

$${}^w \hat{v}_c(k) = u(k) = {}^w v_c(k) + \rho(\mathrm{k}) \tag{19}$$

Where $\rho(k)$ take into account errors that are committed in estimates and dynamics of the system that is not modeled. Substituting an estimated value ${}^w \hat{v}_c(k)$ (eq. (19)) of ${}^w v_c(k)$ in eq. (3):

$$\dot{e}(k) = {}^c v_b^*(k) - {}^c R_w \left({}^w v_b(k) - {}^w v_c(k) - \rho(k) \right) \tag{20}$$

Substituting eq. (7) in (20) we have:

$$\dot{e}(k) = {}^c v_b^*(k) - {}^c R_w \left[{}^c R_w^T \left(\lambda \left({}^c p_b^*(k) - {}^c p_b(k) \right) + {}^c v_b^*(k) \right) \right] - \rho(k) \tag{21}$$

Simplifying:

$$\begin{aligned} \dot{e}(k) &= {}^c v_b^*(k) - {}^c v_b^*(k) + {}^c R_w \rho(k) \\ &\quad - \lambda \left({}^c p_b^*(k) - {}^c p_b(k) \right) \\ &= {}^c R_w \rho(k) - \lambda e(k) \end{aligned} \tag{22}$$

Substituting eq. (20) in our Lyapunov candidate in eq. (17):

$$\dot{V} = e^T(k)\dot{e}(k) = -e^T(k)\lambda e(k) + e^T(k)\, {}^c R_w \rho(k) \tag{23}$$

From above equation, $\dot{V} < 0$ is fulfilled if:

$$\|e\| > \frac{\|\rho\|}{\lambda} \tag{24}$$

Finally if we consider that $\rho(k) \to 0$ then error will tend to zero $e(k) \to 0$ otherwise eq. (24) is not fulfilled error will not decrease but it will be bounded by:

$$\|e\| < \frac{\|\rho\|}{\lambda} \tag{25}$$

We can note that error will increase or decrease depending on error in our estimates. In order to estimate ρ consider that errors of position and velocity estimates are bigger that errors in the dynamics of the system thus $\rho(k)$ can be obtained from equations (7), (8) and (19).

$$\begin{aligned} u(k) &= {}^w \hat{v}_b(k) - {}^c R_w^T \left[\lambda \left({}^c p_b^*(k) - {}^c \hat{p}_b(k) \right) + {}^c v_b^*(k) \right] \\ u(k) &= \rho(k) + {}^w v_b(k) \\ &\quad - {}^c R_w^T \left[\lambda \left({}^c p_b^*(k) - {}^c \hat{p}_b(k) \right) + {}^c v_b^*(k) \right] \end{aligned} \tag{26}$$

And $\rho(k)$ can be expressed as:

$$\rho(k) = {}^w \hat{v}_b(k) - {}^w v_b(k) + {}^c R_w^T \lambda \left({}^c \hat{p}_b(k) - {}^c p_b(k) \right) \tag{27}$$

IV. Results

In this section the controller shown in the eq. (8) is tested and compared with a controller shown in previous works [17]. In order to carry out a isolated comparison

A. Test of the Performance of the Controller

In order to carry out the experiment, the system considers that (for a satisfactory motion of the system in the Cartesian space) translational axes are decoupled, in this work rotational motions are not considered. Spherical object allows to decoupling translational degrees of freedom in a monocular visual system. Thus position of the object in the plane X, Y is obtained from the centroid of the spherical object in the plane of the image and distance in the Z axis is taken from the diameter of the object. On the other hand performance of the controller is especially different along the Z axis. This difference is mainly due to the noise influence in this Cartesian coordinate, as it was mentioned in the section $II{-}E$. For this reason, controller is tested separately in x, y and z coordinates. In order to contrast results obtained from this new controller a previous controller designed for static visual control (in previous works [17]) is compared and subjected to the same test. Previous controller is shown in eq. (28) and its design it is similar to the controller shown in this article.

$$u(k) = {}^w \hat{v}_b(k) - {}^c R_w^T \left[\lambda \left({}^c p_b^*(k) - {}^c \hat{p}_b(k) \right) \right] \tag{28}$$

Controllers are compared by means of a index that isolates de error (28)

$$Tracking\ error\ index = \frac{\|e_{xyz}\|}{\|{}^w \hat{v}_b\|} \tag{29}$$

Where e_{xyz} is the tracking error is vector for the duration of the test and ${}^w \hat{v}_b$ is vector of the estimated velocity of the ball for the duration of the test.

In order to tests the controllers in eq. (8) and (28), by applying index in (29) and with the intention of considering similar conditions, the reference is changing by a function that represents the reference position and velocity (in a real case this function is known since it is possible to design a trajectory to hit or catch the object, in this case a ball).

TABLE I

Error tracking index defined as in (29). Two Kind of Movements of the Ball, where the Ball is moved along each axis. In the last static case, the ball is moved simultaneously along the XYZ axes. In the last dynamic case the ball is moving and the reference being changed along the XYZ axes.

Axis relative displacement	*Tracking err. index* in controller, **eq. (8)**	*Tracking err. index* in controller, **eq. (28)**	**Type of Ball movement**
X	5.795	12.225	Static
Y	5.618	12.035	
Z	5.360	11.742	
X	9.536	15.428	Dynamic
Y	9.488	15.358	
Z	9.679	15.538	
XYZ	7.869	13.826	Static
XYZ	11.956	18.659	Dynamic

Two kind of test were implemented: first the ball was fixed (Static case) and the reference was changing, on the other hand in the second case, the ball was under movement (dynamic case) while the reference was changing. Both controllers were subjected to the same experiment conditions and the same cases. On the other hand, in order to isolate the influence of the kind of movement of the ball over the error, the index in (29) it was applied to all tests. Tests were applied to each axis in a first instance in order to analyze error per axis (due to the differences of the noise influence, especially in the *X* axis -deep estimate-) at the same time as other axes remained relatively constant (6 tests are shown in table 1). Last tests are two additional experiments, where the reference was variable in all axes, in the static case the ball is fixed and in the second the ball was under movement and at the same time the reference is changing. General results are show in table 1 and two experiments are shown in fig. 4. In table 1 is shown that dynamic controller improves error performance almost to the half in all axes and kind of ball movement. In fig. 4 a) and b) difference is notorious, index in table 1 shows that error is drastically reduced from 5.618 to 12.035. In fig. 4 c) and d) error is reduced from 23.324 to 14.942.

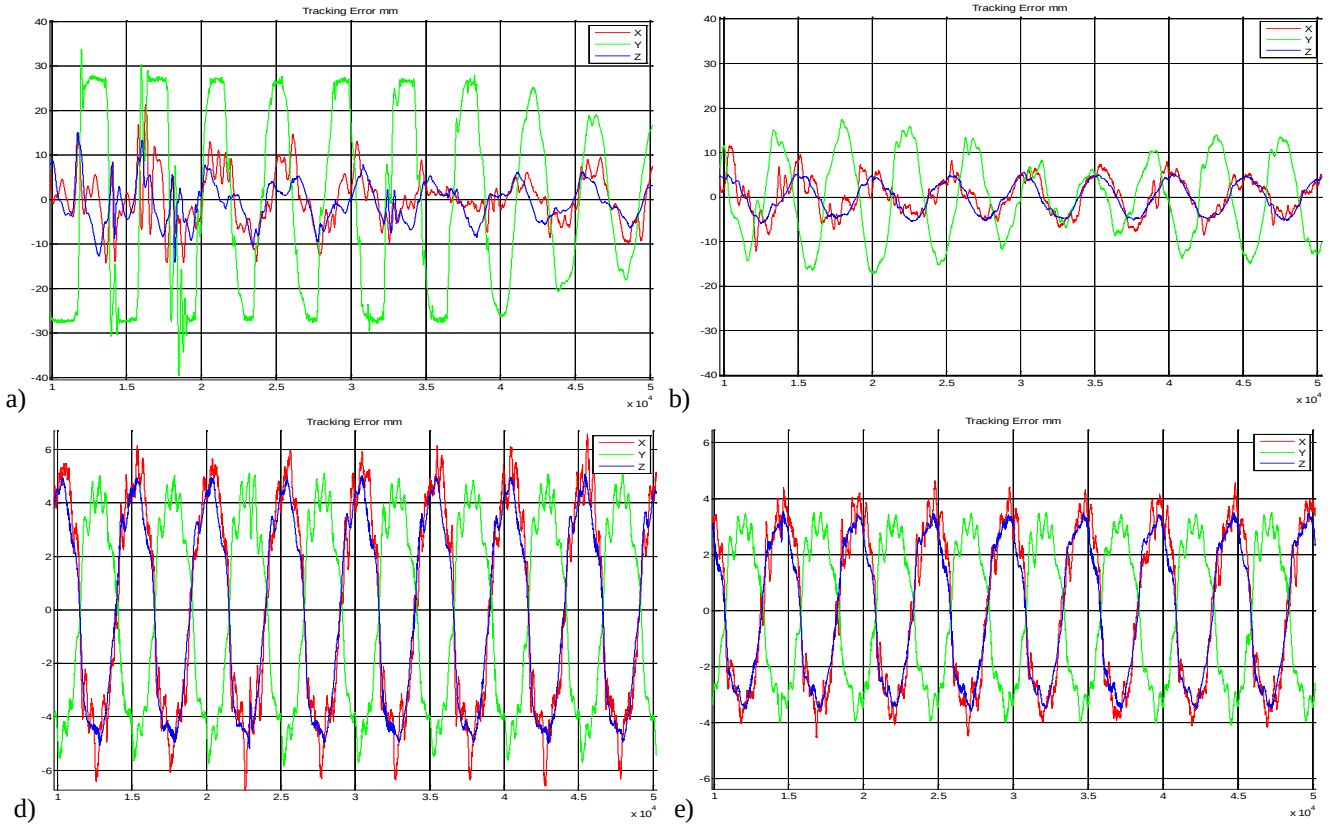

Fig. 4 shows error evolution in the Cartesian space. Figures a) and b) show positioning errors (of two controllers) when the ball is fixed and the reference is changing along the "Y" axis. a) Shows errors that are produced when the controller of eq. (28) is used and the ball is fixed. b) Shows errors that are produced when the controller of eq. (8) is used and the ball is fixed. Figures c) and d) show positioning errors (of two controllers) when ball and reference is changing along XYZ axes simultaneously, c) Shows errors that are produced when the controller of eq. (28) is used and the ball is under movement along three axes simultaneously. d) Shows errors that are produced when the controller of eq. (8) is used and the ball is under movement along three Cartesian axes simultaneously.

B. Game Strategy

Additionally to the results shown above, in order to test the control law in (8), a ping-pong ball game strategy is designed and tested in the robot system. This section it is an invitation to visit the Robotenis web page to see videos and other results.

Basically game strategy consists in hitting the ball in a position that is (estimated) known and inside of the work space of the robot. In order to do this, the workspace area was divided by three planes that are perpendicular to the *Z* axis of the camera. Thus when the ball is "too far" or beyond the plane number 1 (respect to the *Z* axis of the camera) shown in Fig. 5, then the robot tracks the object only in the plane *X* and *Y* (of the camera) in order to guarantee that the object is not too far from the bat of the robot. Second area is contained between the second and first plane, in this area it is possible that the ball may be out of the field of view of the camera and the robot has to decide the point in the where to hit the ball. Thus when the ball is out of the field of view of the camera then the robot is guided by the estimate of the trajectory of the ball. The third area is a zone where the

robot tracks the ball in the plane *XY* of the camera while it plans the trajectory, velocity and the time in which the ball has to be hit (Fig. 5 is complemented in Fig. 6). A video where results are shown can be founded in Robotenis page: http://www.disam.upm.es/vision/projects/robotenis/

CONCLUSIONS

In this paper a visual controller for dynamic tracking was presented. This controller it was designed in order to carry out tasks as can be hitting or caching objects in the 3D space. In this case the object was a black ball in a simplified scene in order to make a fast image analysis. This new visual controller it was compared with a controller designed in a previous work in order to analyze possible advantages and it was observed that error was reduced more than 40% respect to the previous controller. In the web page it can be observed that the controller improves the behavior of the system and the ball when the new control law is included.

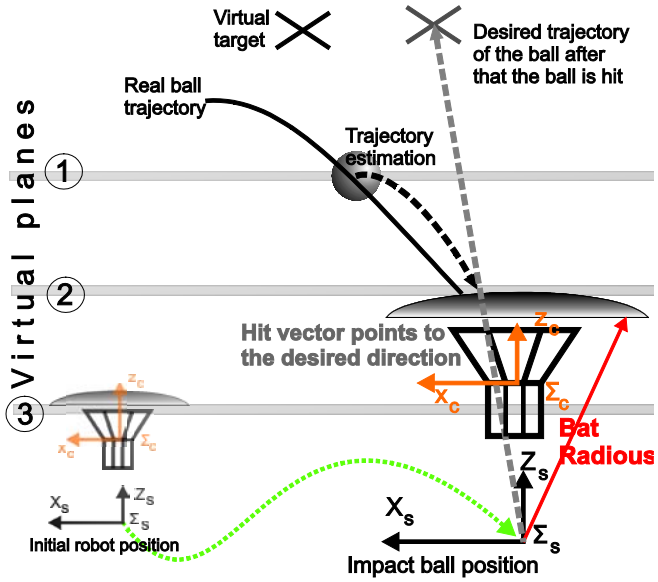

Fig. 5 Basic scheme of the controllers of the system.

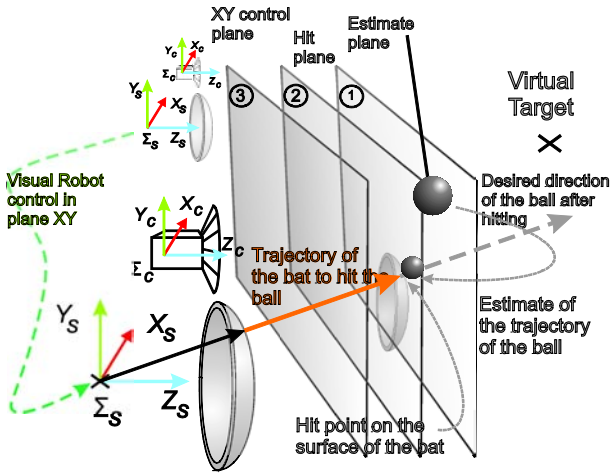

Fig. 6 Basic scheme of the controllers of the system.

ACKNOWLEDGMENT

Authors thanks for financial support to the following institutions: Universidad Politécnica de Madrid from Madrid Spain; PROMEP and Universidad Autónoma de Querétaro from Queretaro, Mexico; and Universidad Nacional de San Juan from San Juan, Argentina.

REFERENCES

[1] D. Kragic and H. Christensen, "Advances in robot vision," *Robotics and Autonomous Systems*, vol. 52, 2005, p. 1–3.

[2] N. Oda, M. Ito, and M. Shibata, "Vision-based motion control for robotic systems," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 4, 2009, p. 176–183.

[3] M. Kaneko, M. Higashimori, R. Takenaka, A. Namiki, and M. Ishikawa, "The 100 G capturing robot- too fast to see," *IEEE/ASME transactions on mechatronics*, vol. 8, 2003, p. 37–44.

[4] T. Senoo, A. Namiki, and M. Ishikawa, "High-speed batting using a multi-jointed manipulator," *IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION*, Citeseer, 2004, p. 1191–1196.

[5] T. Senoo, Y. Yamakawa, S. Mizusawa, A. Namiki, M. Ishikawa, and M. Shimojo, "Skillful manipulation based on high-speed sensory-motor fusion," *Proceedings of the 2009 IEEE international conference on Robotics and Automation*, IEEE Press, 2009, p. 4252–4253.

[6] R. Clavel, "Conception d'un robot parallele rapide a4 degrés de liberté'','' *These de Doctorat*, 1991.

[7] L. Silva, J. Sebastian, R. Saltaren, R. Aracil, and J. Sanpedro, "RoboTenis: optimal design of a parallel robot with high performance," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005.(IROS 2005)*, Ieee, 2005, p. 2134–2139.

[8] L. Ángel, J. Sebastián, R. Saltarén, R. Aracil, and R. Gutiérrez, "RoboTenis: design, dynamic modeling and preliminary control," *2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings*, 2005, p. 747–752.

[9] L. Angel, J. Sebastian, R. Saltaren, and R. Aracil, "Robo Tenis System Part II: Dynamics and Control," *IEEE CONFERENCE ON DECISION AND CONTROL*, IEEE; 1998, 2005, p. 2030.

[10] J. Sebastián, "Robotenis System," *http://www.disam.upm.es/vision/projects/robotenis/*, 2010.

[11] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, 2000, p. 1330–1334.

[12] G. Welch, Greg ; Bishop, "The Kalman Filter. Some tutorials, references, and research related to the Kalman filter" *http://www.cs.unc.edu/~welch/kalman/*, 2010.

[13] A. Traslosheros, J. Sebastián, L. Ángel, F. Roberti, and R. Carelli, "Visual servoing using a parallel robot: Preliminary results," *Advanced intelligent mechatronics*, Zurich: 2007, p. 4–7.

[14] F. Chaumette and S. Hutchinson, *Visual servo control. I. Basic approaches*, 2006.

[15] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, 2007, p. 109–118.

[16] S. Hutchinson, G. Hager, and P. Corke, *A tutorial on visual servo control*, 1996.

[17] L. Angel, A. Traslosheros, J. Sebastian, L. Pari, R. Carelli, and F. Roberti, "Vision-Based Control of the RoboTenis System," *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, vol. 370, 2008, p. 229.

Índice de Autores