Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

Variable Structure PID Based Visual Servoing for Robotic Tracking and Manipulation

Marcello Bonfè[†], Elena Mainardi[†] and Cesare Fantuzzi[‡]

[†] *Università di Ferrara, Dipartimento di Ingegneria, Ferrara, Italy, (mbonfe,emainardi)@ing.unife.it.*
[‡] *Università di Modena e Reggio Emilia, DISMI, Reggio Emilia, Italy, cesare.fantuzzi@ieee.org*

Abstract

This paper presents a practical experiment on the use of visual servoing for robotic tracking and manipulation of objects moving on a plane. In this work, the image is acquired from the camera mounted on the end-effector of a 6 DOF industrial robot, and processed on a standard PC-based vision system in order to close the Visual Servoing loop with a PID based controller. The PID has been implemented with a variable structure, modified according to the estimated velocity of the target, in order to achieve a good step response and to eliminate steady-state error with a moving target. The possibility to perform a stable tracking and to estimate the target dynamics allows also to grasp the object with a prediction based trajectory planning.

1 Introduction

The applications of visual sensors in robot control have had recently a great improvement, also in industrial environments. First applications of artificial vision in robotic systems were concerned with task and path planning in unstructured environments, see f.e. [1], [2]. The typical sequence of operations of these kind of systems can be described as a "look-and-move" strategy, which is in practice an open-loop control. The majority of industrial applications of artificial vision in robotic systems implement this technique, because it can be greatly simplified assuming that the environment is partially known, as is commonly the case in a manufacturing process line. However, as it is easy to argue, improved accuracy of global robot/vision system, can be achieved with the realization of feedback control loops. In this case, the term "visual servoing" is applicable. Visual servoing is now a very common practice in the international robotic academic community, as is exhaustively surveyed by Corke in [3].

In general, the visual servoing problem is to minimize positioning error of the robot (of its end-effector in case of a manipulator) with respect to a desired position, which is identified somehow with the features of the acquired image. The error can be calculated directly in the image space, or estimating the Euclidean position of the robot and then detecting the error in its task space. These two possible approaches are called respectively Image-Based Visual Servoing and Position-Based Visual Servoing. Applications developed with the first approach are described f.e. in [4], [5],while the latter has been used in [8], [7].

Another, more intuitive, distinction that can be made for visual servoing control structures is based on the relative pose between the robot and the camera, which can be *camera-in-hand* or *camera-to-hand*. The first approach gives sharper and simpler information, the second one requires more complex artificial vision algorithms but returns much more information about the complete task environment [10].

In spite of the progress of visual servoing architectures described in the literature, practical implementations have been rarely reported in an industrial framework, due to the complexities in the control and vision systems set-up, which typically need highly specialized hardware, and in the implementation of image processing and visual control algorithms. The experiments described in this paper, instead, represent a visual servoing solution based on commercially "off-the-shelf" components, which can be applied quickly on almost every robot system, without modifying the inner robot control loop provided by the manufacturer–made control system.

The aim of the visual servoing system described in this paper is to track, with the end-effector of a 6 DOF industrial manipulator, a possibly moving object lying on a plane, in order to perform subsequent grasping and manipulation, once that the tracking is stable. The system adopts a *camera-in-hand* approach, and the camera images are acquired and processed by a PC-based vision system in order to determine the displacement between the projection of the robot gripper onto the target's plane XY_t and the target position, that represents the error signal compensated by a PID-based controller. In order to simplify the issues related to camera calibration, the camera/gripper ensemble is virtually constrained to move on a plane parallel to XY_t. This constraint should not be viewed as a dramatical limitation, as a lot of industry–oriented applications, like belt system for material transportation, require a simple bidimensional working area. In any case this practical constraint can be easily removed modifying slightly the procedure for camera calibration.

Our work began with the system modelling phase, during which a simplified dynamic model of the robot and vision system ensemble have been developed, taking into account how the proprietary robot controller could perform a real-time modification of the end-effector position and the latencies introduced by trajectory planning and communication with the PC. It is important to note that the PC–Robot communication link is mono-directional, which means that the information of the actual robot position is not available at visual servo level. The second step concerned the design of a PID controller for the visual-based loop. In order to increase system performances for a possibly varying target dynamics, we have investigated a variable structure for the controller, which, according to an estimate of the target's velocity, changes smoothly from a PD-like behaviour to a PID-like one. This permits to eliminate steady-state error for object's moving at a constant velocity and also to reduce overshoot and settling-time for still targets. Finally, it has been developed a procedure to grasp the tracked object once that the tracking is stable and features of object's motion are proper (e.g. acceleration equals to zero).

The rest of the paper is organized as follows. The hardware and the technology used in the experiments are described in Section 2, while the model of the overall system and the variable structure PID controller are analyzed in Section 3. Experimental results are described in Section 4 and Section 5 gives some concluding remarks.

2 A technological view of the system

The system is composed of a Personal Computer hosting a frame grabber board, a CCD sensors camera and an industrial robot manipulator, connected as shown in Figure 1.

2.1 The robot arm

The manipulator used in the experiments is a Staübli Puma 260 robot, controlled by the Unival Controller Unit ([9]). The arm has 6 DOF, with an operative range of 47 cm. The robot joints are driven by DC permanent magnet motors with incremental encoders feedback.

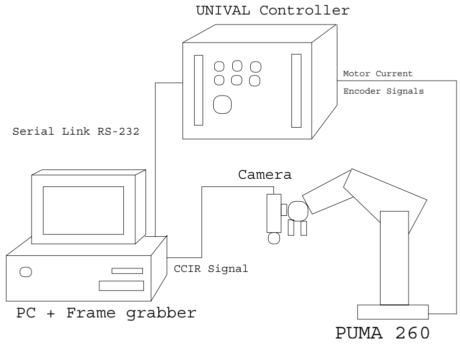

Figure 1: *System connections*

Robot control programs for the Unival Controller, allow real-time modifications of the position to be reached with a movement instruction, according to a correction value that is applied incrementally every system clock tick (32 ms). With this feature, called "Alter Mode", it is possible to realize a sort of velocity control, as described in [6]. In this application, incremental motion commands, coming from the visual controller, are expressed in "tool" coordinates, which means they are referred to the camera focus position.

2.2 The vision system

The vision system is based on a PC host platform, with a 600 MHz Intel PentiumIII processor and 256 Mb of Ram, running Windows NT Operating System without any real-time extensions. The computer communicates with the robot through a standard RS-232 serial link. A Matrox Meteor II frame grabber card and a Jai CV–M10 CCD sensors progressive scan camera, acquiring 30 fps, are the only special purpose components of the system. The image processing routines that have been implemented for this work, extract salient features (area, center of mass, orientation) of every object detected in the image, and heuristics to identify which one has to be tracked can be applied very easily. For the experiments described in this paper, we have chosen to track the center of mass of the object closest to the image center, performing a 2D visual servoing, and to use the information related to object orientation only in the grasping task, which is performed in open-loop, because of the loss of meaning of camera images when the robot gripper approaches the object.

The projection matrix T to transform the task space in the image space is computed thanks to an automatic calibration procedure, in which the robot moves to pre-defined positions while the vision system identifies the position in the image space of a reference target. This procedure allows also to compute, knowing the focus f of the camera, the distance Z_d along a normal direction between the target's plane XY_t and the robot end-effector.

3 Modeling the visual servoing system

The control loop scheme of the eye-in-hand visual servoing architecture implemented in this work can be described as shown in Figure 2. The inner control loop, implemented by the Unival Controller, consists of the trajectory generation, the kinematic inversion and the position control loop.

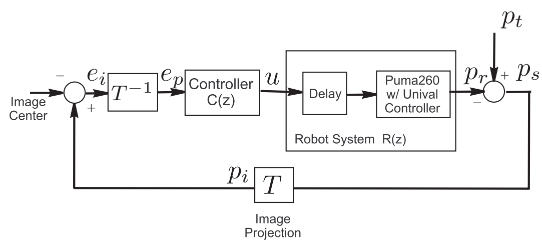

Figure 2: *The visual control scheme*

In the visual control loop, the image acquired by the camera is used to generate the error signal e_i, which is the difference between the position of tracked object's center of mass in the image space, p_i, and the center of the image space, p_0. This error depends from both the target absolute position in the task space, p_t, and the position of the end-effector p_r. Of course, since p_0 is a constant and is the origin of the image space, we can also consider p_t as the input of the system and p_r as the output. In fact, e_i is transformed by the inverse of the projection matrix T, to obtain the displacement in "tool" coordinates, and then processed by the PID-based controller, which determines the incremental motion commands to send to the robot controller.

In order to define a dynamic model of the robotic system, we will assume that the robot structural dynamic effects can be disregarded with respect to visual servo loop timing characteristics, and that each controllable DOF in the planar movement of the end-effector has identical dynamics, so that a single SISO model is valid for both coordinates. Analyzing the realization of real-time path correction with Unival "Alter Mode", which is schematized in Figure 3, we can notice that the input value from the visual servoing controller is periodically added to the current reference position with a sampling time of 32 ms,, after a two-step delay due to trajectory planning and kinematic inversion.

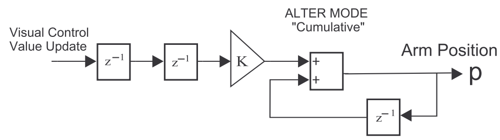

Figure 3: *Model of position dynamics of Puma260 with Unival Controller in "Alter mode"*

Therefore, the transfer function of the robotic ensemble can be expressed as:

$$R(z) = z^{-2}\frac{Kz}{z-1} \tag{1}$$

The gain factor K is a parameter that permits to reduce the effects of great steps in the visual control. Setting $K = 0.1$, resampling the transfer function at the rate of the visual servo loop, which is of 260 ms due to the bandwidth of the communication link, and considering a total processing delay of 260 ms, we obtain:

$$R(z) = \frac{0.7z + 0.1}{z^3 - z^2} \tag{2}$$

3.1 The Variable Structure PID

The vision based controller has been implemented in a first phase of the work, as a simple PID controller. However, this kind of controller is not perfectly suitable to every possible practical situation, especially with regards to the dynamic characteristics of the target. In fact, when the target is still, the best performances can be achieved with a PD structure, because the integral term gives a sensible overshoot in the system response and, of course, an increase in the settling time. On the other hand, if the object is moving with a constant velocity, a PID controller is able to eliminate the steady-state error. From this considerations, it has been studied a control solution based on a variable structure controller that can achieve good performances for both still or moving target tracking.

The implementation of a variable structure PID has been done starting from the classical Z-Transform description of a digital PID:

$$C(z) = \frac{U(z)}{E(z)} = K_p\left(1 + \frac{T}{T_i(z-1)} + \frac{T_d}{T + \frac{T_d}{N}}\frac{z-1}{\left[z - \frac{T_d}{NT+T_d}\right]}\right) \tag{3}$$

which can be reduced to the following difference equation:

$$U_k = U_{k-1}\cdot r_1 - U_{k-2}\cdot r_2 + E_k\cdot q_0 + E_{k-1}\cdot q_1 + E_{k-2}\cdot q_2 \tag{4}$$

assuming settings of Table 1. The structure of this

$q_0 = K_p(1+\beta)$
$q_1 = -K_p(1+\gamma-\alpha+2\beta)$ $\alpha = \frac{T}{T_i}$
$q_2 = K_p(\gamma-\alpha\gamma+\beta)$ $\beta = N \cdot \gamma$
$r_1 = \gamma + 1$ $\gamma = \frac{T_d}{N \cdot T + T_d}$
$r_2 = \gamma$

Table 1: *Parameter settings for PID in difference equation*

generic PID controller can be modified adequately for the purpose of our experiments if we introduce a variable parameter $\Delta \in [0,1]$, and redefine the controller parameters of 4 as:

$$\begin{array}{rcl} q_0 &=& K_p(1+\beta) \\ q_1 &=& -K_p(\Delta+\gamma-\Delta\cdot\alpha+\Delta\cdot\beta+\beta) \\ q_2 &=& K_p\cdot\Delta\cdot(\gamma-\alpha\gamma+\beta) \\ r_1 &=& \gamma+\Delta \\ r_2 &=& \gamma\cdot\Delta \end{array} \quad (5)$$

In fact, for $\Delta = 1$, the controller is exactly a PID and for $\Delta = 0$, it is exactly a PD.

Since we want to modify the controller structure according to the target's dynamics, Δ could be expressed as a function of target's velocity. In order to have smooth variation of the controller parameter, we have chosen a non-linear function that relates Δ to an estimate of the velocity of the target, as shown in Table 2: $\dot{p}_{sw}$ is the threshold between the PD-like behaviour and the PID-like behaviour and n permits to modify the speed of change from 0 to 1 of Δ.

$$\Delta(\dot{\hat{p}}_t) = \frac{\left(\frac{\dot{\hat{p}}_t}{\dot{p}_{sw}}\right)^{2n}}{\left(\frac{\dot{\hat{p}}_t}{\dot{p}_{sw}}\right)^{2n}+1}$$

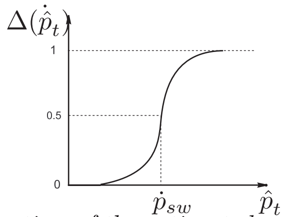

Table 2: *Δ as a non-linear function of the estimated velocity of the target.*

As stated in Section 1, at visual servo level the only information about robot and target position comes from camera sensor measurement, which gives the relative displacement in a certain planar projection. This also means that it is not possible to separate the movement of target's projection in the image space into the contribution of robot motion and target's own motion, without computing a dynamic estimate of the robot position.

In fact, as shown in Figure 4, an observer that implements the model in 2, is used to estimate the end-effector absolute position $\hat{p}_r$. Then, the position of the target's center of mass in the image space, p_i, is transformed by means of matrix T^{-1} in task space coordinates, recovering the task space displacement p_e. The result of adding $\hat{p}_r$ to p_e, is an estimate of the target's absolute position $\hat{p}_t$, whose time derivative can be considered as an estimate of the target's velocity. The derivative block has been implemented as a filtered derivative, in order to reduce noise and quantization effects. The value of $\dot{\hat{p}}_t$ is exactly what

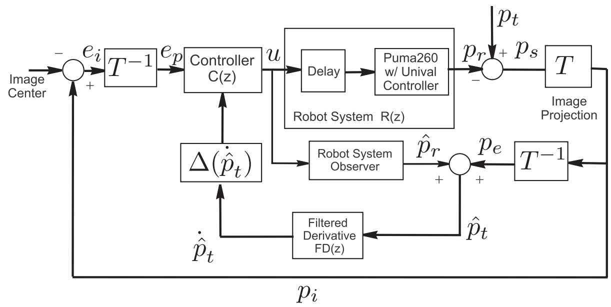

Figure 4: *The complete control system block diagram*

we need as the argument of the function that express the adaptation of the controller parameter to the dynamic features of the target.

3.2 Stability analysis

The constant parameters of the Variable Structure PID described in previous section, have been tuned with classical methods (e.g. to achieve a sufficient phase margin, to place poles in a stable region), considering the most critical case with regard to the stability of the closed-loop system, which is that of a pure PID behaviour ($\Delta = 1$). In particular, the following settings have been applied: $K_p = 0.3$, $T_i = 1.5$, $T_d = 0.1$ and $N = 10$. However, this is not sufficient to ensure that the global parameter-varying system is stable. For that purpose, it is possible to apply a quadratic stability test expressed with Linear Matrix Inequalities (LMI) [11], a novel theory that has found applications in stability proof for Fuzzy Systems [13] and Robust Control [14]. In particular, the following lemma:

Lemma 1 *[14] The system $x(k+1) = \mathbf{A}(\alpha)x(k)$ where the dynamic matrix $\mathbf{A}(\alpha)$ belongs to a convex polytopic set defined as $\mathcal{A} := \left\{\mathbf{A}(\alpha) : \mathbf{A}(\alpha) = \sum_{i=1}^{N}\alpha_i\mathbf{A}_i, \sum_{i=1}^{N}\alpha_i = 1, \alpha_i \geq 0\right\}$ is quadratically stable if there exist a symmetric positive definite matrix $\mathbf{P} > 0$ such that:*

$$\mathbf{A}_i^T\mathbf{P}\mathbf{A}_i - \mathbf{P} < 0 \quad (6)$$

for $i = 1, ..., N$.

can be applied in our case if we consider that the closed-loop system dynamics, without inputs, can be expressed in state-space as:

$$\begin{aligned} x(k+1) &= \mathbf{A}(\Delta)x(k) \\ \mathbf{A}(\Delta) &= \Delta\mathbf{A}_1 + (1-\Delta)\mathbf{A}_2 \end{aligned} \tag{7}$$

where $\mathbf{A}_1$ corresponds to the closed-loop system with a PID controller, and $\mathbf{A}_2$ to the closed-loop system with a PD. Lemma 1 gives a sufficient condition to prove numerically the stability of the free visual servoing loop, finding a feasible solution to the LMI problem 6. The solution has been found with the help of Matlab® and the `LMItool` of El Ghaoui and Boyd [12], and it is:

$$\mathbf{A_1} = \begin{pmatrix} 2.037 & -1.3617 & 0.3327 & -0.0359 & -0.012 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{A_2} = \begin{pmatrix} 1.037 & -0.3247 & 0.0444 & 0.0122 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\mathbf{P} = 10^3 \cdot \begin{pmatrix} 1.5047 & -1.8809 & 0.6105 & 0.0004 & -0.0124 \\ -1.8809 & 2.8424 & -1.3482 & 0.1466 & 0.0506 \\ 0.6105 & -1.3482 & 1.2683 & -0.5421 & 0.0499 \\ 0.0004 & 0.1466 & -0.5421 & 0.6312 & -02329 \\ -0.0124 & 0.0506 & 0.0499 & -0.2329 & 0.1688 \end{pmatrix}$$

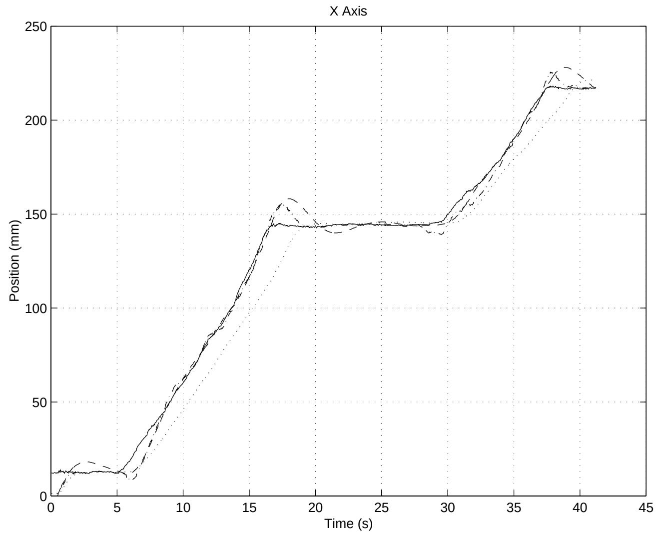

Figure 5: *Tracking in X axis position (solid line: target; dashed: visual servo with PID; dotted: visual servo with PD; dash-dotted: visual servo with VSPID).*

4 Experimental results

In Figure 5 and Figure 6 are reported the tracking results of an experiment in which the target was initially still for 5 s, then started to move for 7 s at about 1 cm/s parallel to the X axis, changed direction and moved parallel to Y axis for 16 s and finally changed direction again to move parallel to X axis for 8 s more.

The test permits to compare the Variable Structure PID controller with classical controllers with the same parameters but fixed PD and PID structure.

Numerical results say that the peak-to-peak tracking error with respectively a PID, PD and VSPID controller are: $e_{pp}^{PID} = 14.86$mm, $e_{pp}^{PD} = 16.45$mm, $e_{pp}^{VSPID} = 13.6$mm; while the errors' RMS values are: $e_{rms}^{PID} = 5.02$mm, $e_{rms}^{PD} = 7.5$mm, $e_{rms}^{VSPID} = 4.2$mm. Moreover, from the analysis of the graphical comparison, the following remarks can be done:

1. when the target is initially still, the settling time of the VSPID is the same of the PD controller, which is in fact better than the classical PID.

2. when the target starts moving, while the classical PD has a steady-state error, the VSPID can eliminate it, as the normal PID does.

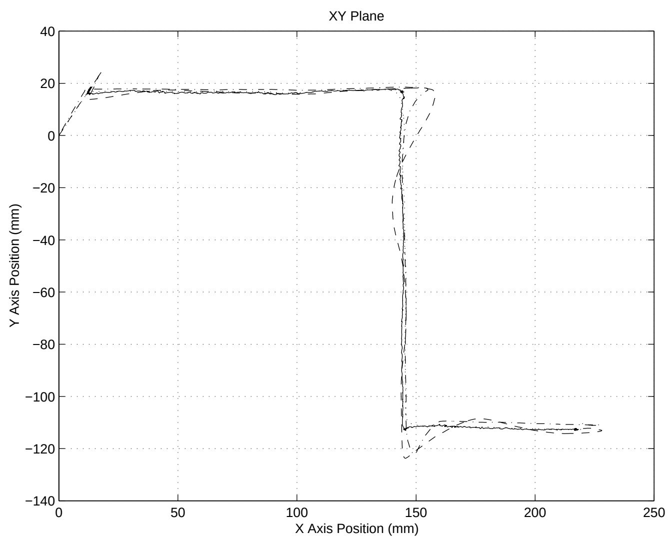

Figure 6: *Tracking in the XY plane (solid line: target; dashed: visual servo with PID; dotted: visual servo with PD; dash-dotted: visual servo with VSPID).*

3. when the target stops after moving for a while, the VSPID, as it changes its structure when detecting the stop, is able to reduce overshoot and settling time, that is instead considerable with the classical PID.

To conclude, the VSPID visual servoing loop is able to track a moving target better than a classical PID controller, especially when the target can have pretty different dynamics in different time intervals.

Since with the VSPID is possible to perform a stable tracking of a moving object and since an estimate of target dynamics is available, it is also possible to realize a procedure for target grasping based on a prediction of the future position of the object at a certain time, and in the computation of an approaching trajectory for the robot end-effector. In the simplest case, the approach and grasping operations can be performed when the tracking error is stabilized under a certain value and the object's velocity is constant in both directions (linear planar trajectory). The following information are necessary to perform the trajectory planning: the desired time to perform the grasp T_d; the distance Z_d, along a normal direction, between the object's plane and the end-effector virtual planar constraint, computed automatically by the calibration procedure described in 2.2; the orientation θ_o of the object in the image space, which is needed because the robot gripper is two-fingered; the distance between the gripper and camera focus d_{gf}; the target's velocity in X and Y directions $\hat{v}_X^o$, $\hat{v}_Y^o$. The point to reach to grasp the object is then (in "tool" coordinates"):

$$\begin{array}{rcl} X_f & = & T_d\hat{v}_X^o + d_{gf}cos\theta \\ Y_f & = & T_d\hat{v}_Y^o + d_{gf}sen\theta \\ Z_f & = & Z_d \\ O_Z & = & \theta_o \end{array} \quad (8)$$

that requires a straight line movement with linear velocity $v_r = \frac{\sqrt{X_f^2+Y_f^2+Z_f^2}}{T_d}$.

The experiments performed proved that the object's can be effectively tracked and grasped, even if the final grasping trajectory is executed in open-loop and is based only on estimated information.

5 Conclusion

The paper concerned with experiments on visual servoing for objects tracking and manipulation with a system based on a commercial robot and a PC–hosted vision system. The PC performs the computation tasks needed for the extraction of image features of a tracked object and for the the generation of the visual–based tracking trajectory, while the robot controller performs the kinematic inversion and the position control loop. The visual servoing regulator is based on classical PID scheme, but the enhancement of a variable structure depending on the estimated velocity of the target, has been proposed to improve tracking performances.

References

[1] Lazano-Perez T., Jones J.L., Mazer E, et al. "Handey: a robot system that recognizes, plans and manipulates", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1987.

[2] McCain H.G., "A hierarchically controlled, sensory interactive robot in the automated manufacturing research facility", *Proc. IEEE Int.Conf. on Robotics and Automation*, 1985.

[3] Corke P.I., "Visual control of robot manipulators - a review", in *"Visual Servoing"* (Hashimoto K., ed.), Robotics and Automated Systems Vol.7, World Scientific, 1993.

[4] Feddema J.T., Mitchell O.R. "Vision-guided servoing with feature-based trajectory generation", *IEEE Trans. Robotics and Automation*, Vol. 5, Issue 5, Oct. 1989.

[5] Hashimoto K., Kimoto. T, Ebine T., Kimura H., "Manipulator control with image-based visual servo", *Proc. IEEE Int. Conf. on Robotics and Automation*, 1991.

[6] Corke P.I., Good M.C., "Dynamic effects in visual closed-loop systems", *IEEE Trans. On Robotic and Automation*, Vol. 12, Issue: 5 , Oct. 1996.

[7] Corke P.I., Good M.C., "Controller design for high-performance visual servoing", *Proc. of IFAC World Congress*, 1993.

[8] Papanikolopoulos N., Khosla P., " Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision", *IEEE Trans. Robotics and Automation*, Vol. 9, Issue 1, Feb. 1993.

[9] *"Users's Guide to Robot PUMA 260"*, Unival Robot Controller Arm Manual, June 1993.

[10] Flandin G., Chaumette F., Marchand E., "Eye-in-hand / Eye-to-hand Cooperation for Visual Servoing", *Proc. IEEE Int. Conf. Robotics and Automation*, 2000.

[11] Boyd S., El Ghaoui L., Feron E., Balakrishnan V. *"Linear Matrix Inequalities in System and Control Theory"*, SIAM, 1994.

[12] El Ghaoui L., Nikoukhah R., Delebecque F. "LMI-TOOL: a package for LMI Optimization", *Proc. IEEE Conf. on Decision and Control*, 1995.

[13] Tanaka K., Sugeno M., "Stability analysis and design of fuzzy control systems", *Fuzzy Sets and Systems*, Vol. 45, Issue 2, Elsevier Science, 1992.

[14] De Oliveira M.C., Bernussou J., Geromel J.C., "A new discrete-time robust stability condition", *Systems and Control Letters*, Vol. 37, Issue 4, Elsevier Science, July 1999.