# Non-Linear PID Control for Robotic Manipulator Visual Servoing

M. Bonfè[†], D. Fabbri[†] and C. Fantuzzi[‡]
[†] Università di Ferrara
Dipartimento di Ingegneria
via Saragat, 1 44100, Ferrara, Italy.
[‡] Università di Modena e Reggio Emilia, DISMI
via Allegri 15, Reggio Emilia, Italy.

*Abstract*— **This paper presents an experimental study on the use of visual servoing for robotic tracking and manipulation of targets 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, so that the camera moves on a plane parallel to that of object motion. Using a PID visual controller to minimize the relative displacement between object and end-effector, it can be seen that performances in system response are affected by the absolute velocity of target in space. In order to improve these performances, a non-linear structure for the PID controller has been implemented. The idea is to modify the PID structure according to the estimated velocity of the target. Experiments show that the modified PID controller increase tracking performances for both slow or still targets and fast moving targets.**

*Keywords*—**Visual servoing, PID Control, Target Tracking.**

## I. Introduction

THE applications of visual sensors in robot control have had recently a great improvement, also in industrial environments, thanks to technological enhancements of computer systems and increasing efficiency of artificial vision algorithms, so that it is now possible to obtain an extremely high ratio between the knowledge quantity extracted from vision sensors signals and their cost.

First robotic systems exploiting artificial vision where trying to perform 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 a sort of 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, a good alternative in order to improve accuracy of global robot/vision system, can be the realization of feedback control loops. In this case, the term "visual servoing" is applicable. In fact, this terminology has been introduced exactly to distinguish feedback control based on visual information from "look-and-move" applications. Visual servoing is now a very common term in the international robotic community. An exaustive list of the fundamental visual servoing works is given by Peter Corke in [3].

An important distinction that can be made in visual servoing techniques, regards how the error is computed from the measurement and, according to that, how the control law is determined. In fact, in visual servoing problems one would like to minimize positioning error of the robot (of its end-effector in case of a manipulator) with respect to a desired position, identifying it somehow with the features of the acquired image. The robot task space is mapped into the image space, by means of the projective geometry, according to the transformation:

$$\mathbf{T} : \mathrm{T} \rightarrow \mathrm{F}$$

However, the error can be calculated directly in the image space, or estimating the Cartesian position of the robot and then detecting the error in its task space. This two possible approaches are called respectively Image-Based Visual Servoing and Position-Based Visual Servoing. In Image-Based applications, see f.e. [4] and [5], it is normally necessary to determine the Image Jacobian, which is the matrix that transforms robot end-effector velocity in the task space and interesting features velocity in the image space:

The visual control law is typically based on the computation of the interesting features error in the image space and on the on the inverse (or pseudo-inverse) matrix of the Image Jacobian. As the Image Jacobian is in general depending on the robot position and possibly affected by singularities, it is evident that the major problem in Image-based approach is the inverse operation to be applied on Image Jacobian. However, many authors recall its advantage in terms of robustness with respect to uncertainties in camera intrinsic parameters and in knowledge of its relative pose with respect to the robot.

In Position-Based Visual Servoing, instead, the major problem is represented by the transformation into the tridimensional Cartesian space of an information decoded in one or more (if stereo vision is used) bidimensional spaces. In this case, in fact, the control system is much more sensible to parametric uncertainties related to this transformation. However, once that the control problem is reduced to the kinematic field, it is possible to solve it with classical control schemes, e.g. PID, LQG, Predictive Control, and so on. For example, a comparison between some of these methods applied to visual servoing is described in [7].

Another, more intuitive, distinguishment that can be made for visual servoing control structures is based on the relative pose between the robot and the camera, that leads to the *eye-in-hand* and *eye-to-hand* terminology. The first approach gives sharper and simpler information, the second one requires more complex artificial vision algorithms but returns much more information about the surrounding environment [13].

The aim of the visual servoing system described in this paper is to track with the robot end-effector a possibly moving object lying on a plane, in order to perform subsequent grasping and manipulation, once that the tracking is stable. The system, shown in Figure 1, uses an *eye-in-hand* approach for the control of a 6 DOF industrial robot manipulator. The image acquired from the camera is used to determine the erroneous displacement between robot and target, so that we can consider the visual servoing loop a position-based control. In order to simplify the issues related to camera calibration, the camera/end-effector ensemble is virtually constrained to move on a plane parallel to $XY_t$, the planar physical contraint for object's movement. 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 software for camera calibration.

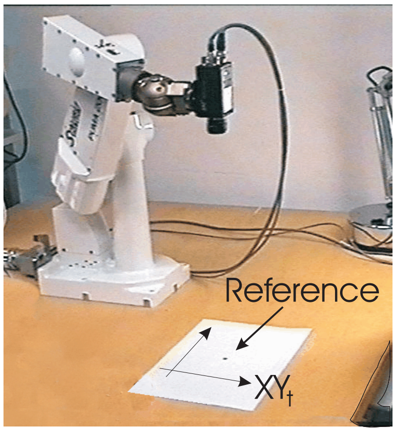


Fig. 1. The experimental set–up

Unlike other experiments on visual servoing, we did not modify the inner robot control loop, keeping the manufacturer–made control system. This has been done mainly because we wanted to investigate a visual servoing solution based on ready "off-the-shelf" components, which can be applied quickly on almost every robot system. On the other hand, in this situation the control platform is composed by two physically separated sub-systems that have to exchange data with some communication link. The bandwidth of the communication link is then a limit for the outer control loop bandwidth, and it also introduce delays in the overall system. Because the bandwidth of the specific visual loop described in this paper is significantly lower than that of robot controller, the robot complex dynamics can be disregarded in the definition of the visual controller.

In fact, a common characteristic of most of previous works on visual servoing is that the robot manipulator is assumed to be a positioning device whose dynamics do not interact with the visual feedback loop. Although this assumption is valid for slow robot motion, it does not hold for high–speed tasks [14]. This becomes a big challenging problem especially when it is not possible to modify the robot joint-level control hardware nor we have deep knowledge on its internal functioning, as it is in our case.

Our work begun with the system modelling phase, in which we developed a simple dynamic model of the robot and vision system ensemble, including the proprietary robot controller behaviour as a black box. The second step concerned the design of a PID controller for the visual-based loop. In order to increase system performances for a wide variety of target dynamics, we have investigated a non-linear structure for the controller, which, according to the estimated velocity of the target determined from a robot model-based observer, switches smoothly from a PD-like behaviour to a PID-like one.

The rest of the paper is organized as follows. The system and the technology used in the experiments is described in Section II, while the model of the overall system, which includes the robot and the vision subsystems, is shown in Section III. The structure of the PID controller and the experimental results are described in Section IV. Section V reports about some concluding remarks.

## II. 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 2.

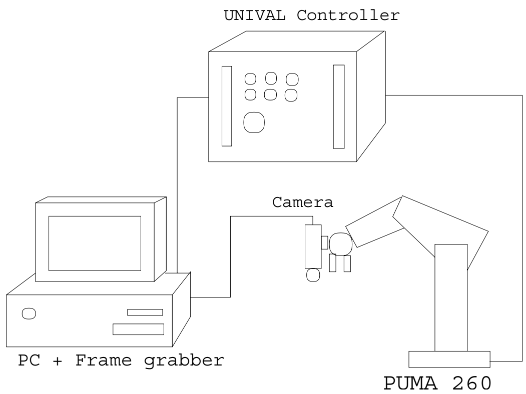


Fig. 2. System connections

### A. The robot arm

The manipulator used in the experiments is a Staübli Puma 260 robot, controlled by the Unival Controller Unit ([10], [11]). The arm has 6 DOF, with an operative range of 47 cm. The robot joints are driven by DC permanent magnet motors and the control system realizes the position and speed control loops for each joint using 6 incremental encoders. Figure 3 shows the Puma 260 robot and its main coordinate modes, called respectively "world" and "tool".

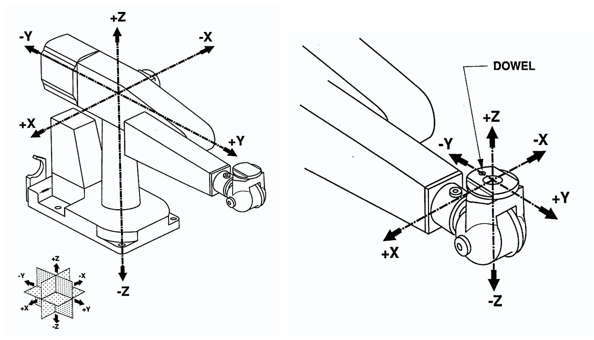


Fig. 3. "World" and "Tool" coordinates modes

Robot control program for the Unival Controller written for this application, performs an endless cycle of straight line movements towards a reference position. This position is corrected in real-time thanks to a specific operative mode, called "Alter Mode", of the Unival controller. With this feature, a parallel task increments the "tool" coordinates of the reference point according to the control values coming from the PC-hosted visual servoing loop.

### *B. The Personal Computer platform*

The host PC is equipped with a 600 MHz Intel PentiumIII processor, 256 Mbyte of Ram and a 9200 rpm hard disk. The computer communicates with the robot through a serial link, and a specific software which interacts directly with the Unival unit control program, as described above, has been written to control the input–output data streaming.

### *C. The vision system*

The vision system is based on a Matrox Meteor II frame grabber card and a Jai CV–M10 CCD sensors progressive scan camera, which can operate at a sample frequency of 30 Hz. Progressive scan cameras are the most proper type of cameras for this kind of applications, as they allow to take pictures of a moving object neater than we could obtain with a common interlaced camera. This is due to the different way of image acquisition adopted by the two types of cameras: interlaced cameras scan odd and even image pixels rows at different times. This means that, if an object is moving, his position will change between the scanning of odd and even fields. The faster the object moves, the more the image will be blurred. This doesn't happen with progressive cameras because they capture all the pixel rows at the same time.

The image processing routines that have been implemented specifically for this work, extract salient features (area, center of mass, orientation) of every objects detected in the image, and heuristics to identify the object to be tracked can be applied very easily. For the experiments described in this paper, we have choosen to track only the center of mass of the object closest to the image center, performing a 2D visual servoing.

## III. Modeling of 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 4. The inner control loop, implemented by the Unival Controller, consists of the trajectory generator, the inverse kinematic solver and the Cartesian position control loop.

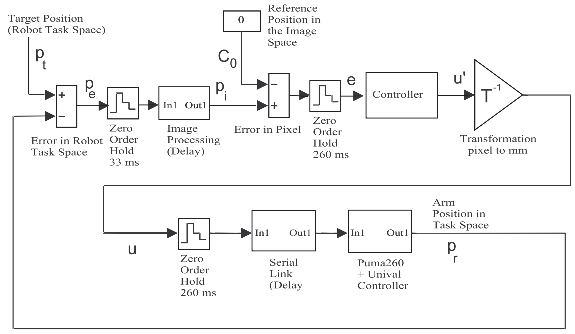


Fig. 4. The visual control scheme

In the outer controller, the actual image acquired by the camera is used to generate the error signal, which is the difference between the the position of tracked object's center of mass in the image space, $p_i$, is compared to the central position of the image space, $C_0$. This error depends from both the target absolute position in the task space, $p_t$, and the position of the end-effector $p_r$. The error is then processed by the PID-based controller, which determines the control value representing an incremental correction to the actual robot position in Cartesian space. The matrix $T^{-1}$ realize the necessary transformation between image space and robot task space.

An important thing to note, is that only from the analysis of the sensor measurement it is not possible to reconstruct the absolute position of the robot end-effector and of the target, but only the relative displacement in a certain planar projection. This also means that it is not possible to determine if the object is still or moving. In order to extract these informations from the image acquired at a certain time, it is necessary to have a dynamic estimate of the robot position.

A model-based observer for the robotic system used in this application needs a deeper investigation of how the Unival controller act in response to an input value from the visual servoing loop. As described in previous section, the robot controller operates in the so–called "Alter mode": this means that the correction values are cyclically accumulated to the actual position every 32 ms. However, the sampling time of the outer control loop is 260 ms and the control input is applied after a delay of about 320 ms. For this reason, the absolute position displacement sent to the robot is scaled of a gain factor K = 0.1, so that the correct position can be reached ideally after 320 ms. Because 32 ms is the shortest sampling time in the overall system, we can choose it as the time basis in the definition of the model.

The robot system working in "Alter mode" can be modeled as schematized in Figure 5, which shows a two-step delay due to the trajectory planning and inverse kinematic solution computation.

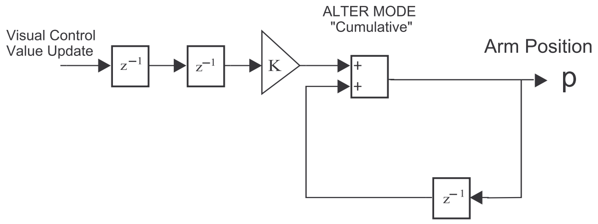


Fig. 5. Model of position dynamics of Puma260 with Unival Controller in "Alter mode"

With regards to the delay introduced by the serial link, it has been calculated as a value of 260 ms. This means that, with the 32 ms sampling time, it can be modeled as an eight-step delay with a reasonable level of confidence.

Assuming that each controlled DOF has identical dynamics, we can define a SISO model for each direction of planar movements of the robot end-effector. A linear state-space model for the robot system dynamics, is the following:

$$\begin{cases} \vec{x}_{k+1} & = & \mathbf{A} \cdot \vec{x}_k + \mathbf{B} \cdot \vec{u}_k \\ \vec{p}_k & = & \mathbf{C} \cdot \vec{x}_k + \mathbf{D} \cdot \vec{u}_k \end{cases} \quad (1)$$

where, because of the overall delay of ten samples, the state vector has ten elements. The matrix $\mathbf{A}$ is then $10 \times 10$, even though its most significative row is the $10^{th}$ one, which represents the effective actuation of the position corrective value, as shown in Figure 5. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are defined as follows:

$$\mathbf{A} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&K&1 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&1 \end{pmatrix} \qquad \mathbf{D} = \begin{pmatrix} 0 \end{pmatrix}$$

As said above, this model can be used to estimate the actual position of the robot end-effector and then calculate the contribution of target's own movement to the position of its center of mass in the image space. In this way, it is possible to evaluate if the object is still or effectively moving.

## IV. Control system and experimental results

The visual control loop described in previous section, 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 absolute velocity of the target. In fact, when the target is still or moving very slow, the best performances in target tracking can be achieved with a PD structure, because the integral term determines 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, the PID controller is able to eliminate the steady-state error.

From this considerations, it has been studied a control solution that could have the best performances obtainable with a PID-like regulator with both still or moving targets. The proposed solution is based on a non-linear controller that changes its structure according to estimated velocity of the target. This estimate is obtained as described in Figure 6

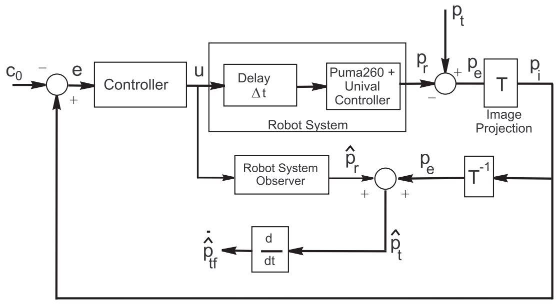


Fig. 6. Block diagram of the target's velocity estimate

As can be seen, an observer that implements the model described in Section III, is used to estimate the robot 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 is needed to determine when it is necessary to switch from a PD controller to a PID, in order to compensate the target's velocity leading to a steady-state error.

The implementation of a non-linear PID has been done starting from the classical Z-Transform description of a digital PID controller:

$$D(z) = \frac{U(z)}{E(z)} = K_p \left( 1 + \frac{T}{T_i(z-1)} + \frac{T_d}{T + \frac{T_d}{N}} \frac{z-1}{\left[ z - \frac{T_d}{NT+T_d} \right]} \right) \quad (2)$$

which can be reduced to the following difference equation:

$$U_k = U_{k-1} \cdot r_1 - U_{k-2} \cdot r_2 + E_k \cdot q_0 + E_{k-1} \cdot q_1 + E_{k-2} \cdot q_2 \quad (3)$$

if we consider that:

$$\alpha = \frac{T}{T_i}$$
$$\beta = N \cdot \gamma$$
$$\gamma = \frac{T_d}{N \cdot T + T_d}$$

and:

$$q_0 = K_p(1+\beta)$$
$$q_1 = -K_p(1+\gamma-\alpha+2\beta)$$
$$q_2 = K_p(\gamma-\alpha\gamma+\beta)$$
$$r_1 = \gamma+1$$
$$r_2 = \gamma$$

The structure of this generic PID controller can be modified adequately for the purpose of our experiments if we introduce a correction parameter $\Delta$ that multiplies opportunely some of the parameters described above. In particular, if we consider $\Delta \in [0,1]$, we can express the parameters of 3 as:

$$\begin{aligned} q_0 &= K_p(1+\beta) \\ q_1 &= -K_p(\Delta+\gamma-\Delta\cdot\alpha+\Delta\cdot\beta+\beta) \\ q_2 &= K_p\cdot\Delta\cdot(\gamma-\alpha\gamma+\beta) \\ r_1 &= \gamma+\Delta \\ r_2 &= \gamma\cdot\Delta \end{aligned} \quad (4)$$

In fact, for $\Delta = 1$, the controller is exactly a PID and for $\Delta = 0$, it is exactly a PD. If we can express $\Delta$ as an adequate function of $\dot{\hat{p}}_t$, we have finally the implementation of a non-linear PID that changes structure according to the target's velocity.

The non-linear function of $\dot{\hat{p}}_t$ we have choosed, shown in Figure 7, is the following:

$$\Delta(\dot{\hat{p}}_t) = \frac{\left(\frac{\dot{\hat{p}}_t}{\dot{p}_{sw}}\right)^{2n}}{\left(\frac{\dot{\hat{p}}_t}{\dot{p}_{sw}}\right)^{2n}+1} \quad (5)$$

where $\dot{p_{sw}}$ is the threshold between the PD-like behaviour and the PID-like behaviour.

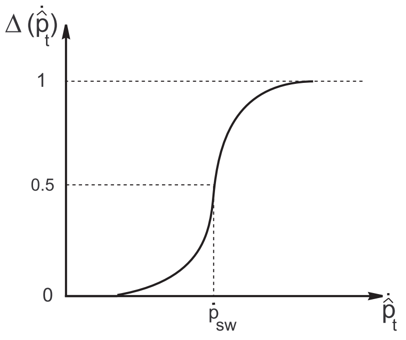


Fig. 7. Correction factor $\Delta$ as a non-linear function of target's velocity

The complete scheme of the control system implemented is then described as shown in Figure 8.

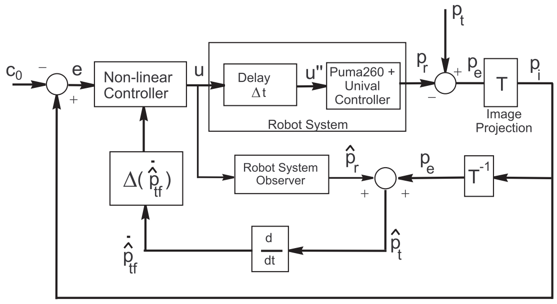


Fig. 8. The complete control system block diagram

In Figure 9 is shown a comparison between system responses to the same target trajectory of the non-linear PID controller and classical controllers with the same parameters but with fixed PD and PID structure.

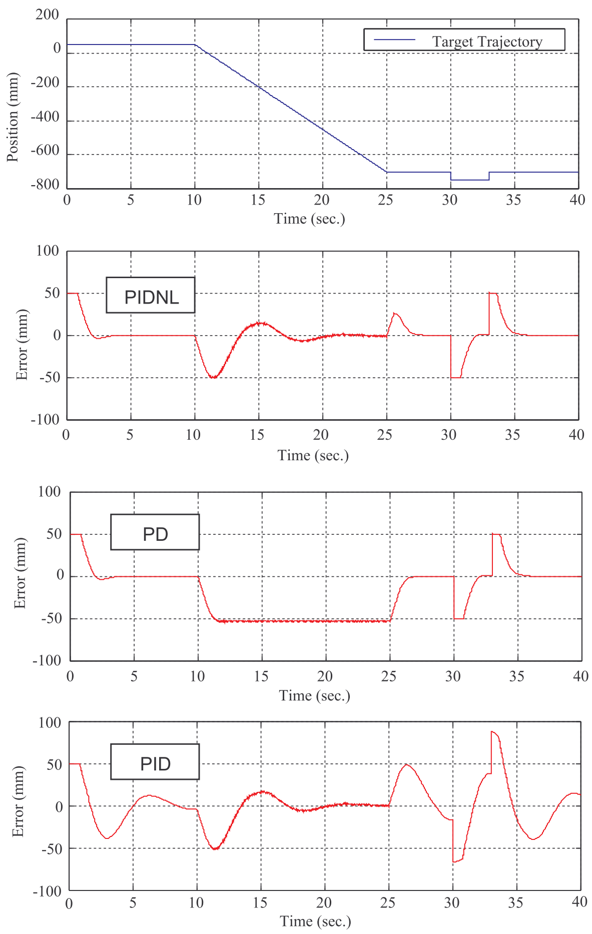


Fig. 9. Comparison between classical controllers and non-linear PID

From the analysis of these results, the following remarks can be done:

1. when target is initially still, the settling time of the non-linear PID 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 non-linear PID can eliminate it, as does the normal PID.

3. when the target stops after moving for a while, the non-linear PID, as it changes its structure in response to the stop detection, is able to reduce the overshoot, and then the settling time, that is instead considerable with the classical PID.

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

## V. Conclusion

The paper concerned with experiments on visual servoing 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 inner position control loop. The visual servoing regulator is based on a classical PID, but some enhancements, such as a non–linear structure that changes according to the estimated velocity of the target, has been proposed to optimize the control performance.

## 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 Conf. 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] 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.

[8] Malis E., Chaumette F., Boudet S., "2-1/2-d visual servoing," IEEE Trans. on Robotics and Automation, vol. 15, Apr. 1999.

[9] McKerrow P. J. "*Introduction to robotics*". Addison-Wesley – 1994.

[10] "*Users's Guide to VAL II,* ", Unival Robot Controller Programming Manual, Dicember 1993.

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

[12] Corke P I., Hutchinson S.A., "*Real-Time Vision, Tracking and Control*". Proc. IEEE Int. Conf. Robotics and Automation 2000.

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

[14] Kelly R., Carelli R., Nasisi O., Kuchen B., Reyes F., "*Stable visual servoing of camera–in–hand robotic systems*". IEEE/ASME Transactions on Mechatronics, vol.5, Issue 1, March 2000.

[15] Ogata K., "*Modern control engineering*". Prentice Hall International Editions – 1990.

[16] Aström K. and Hägglund T., "*PID Controllers: Theory, Design, and Tuning*". Instrument Society of America, 1995.