Hindawi Publishing Corporation
Journal of Robotics
Volume 2012, Article ID 103954, 11 pages
doi:10.1155/2012/103954

Research Article

A Comparison between Position-Based and Image-Based Dynamic Visual Servoings in the Control of a Translating Parallel Manipulator

G. Palmieri,[1] M. Palpacelli,[2] M. Battistelli,[2] and M. Callegari[2]

[1] *Facoltà di Ingegneria, Università degli Studi e-Campus, 22060 Novedrate, Italy*
[2] *Dipartimento di Ingegneria Industriale e Scienze Matematiche, Università Politecnica delle Marche, 60121 Ancona, Italy*

Correspondence should be addressed to G. Palmieri, giacomo.palmieri@uniecampus.it

Received 9 November 2011; Accepted 31 January 2012

Academic Editor: Huosheng Hu

Two different visual servoing controls have been developed to govern a translating parallel manipulator with an eye-in-hand configuration, That is, a position-based and an image-based controller. The robot must be able to reach and grasp a target randomly positioned in the workspace; the control must be adaptive to compensate motions of the target in the 3D space. The trajectory planning strategy ensures the continuity of the velocity vector for both PBVS and IBVS controls, whereas a replanning event is needed. A comparison between the two approaches is given in terms of accuracy, fastness, and stability in relation to the robot peculiar characteristics.

1. Introduction

Visual servoing is the use of computer vision to control the motion of a robot; two basic approaches can be identified [1–4]: *position-based visual servo* (PBVS), in which vision data are used to reconstruct the 3D pose of the robot and a kinematic error is generated in the Cartesian space and mapped to actuators commands [5–7]; *image-based visual servo* (IBVS), in which the error is generated directly from image plane features [8–15]. Recently, a new family of hybrid or partitioned methods is growing, with the aim of combining advantages of PBVS and IBVS while trying to avoid their shortcomings [16, 17].

The principal advantage of using position-based control is the chance of defining tasks in a standard Cartesian frame. On the other hand, the control law strongly depends on the optical parameters of the vision system and can become widely sensitive to calibration errors. On the contrary, the image-based control is less sensitive to calibration errors; however, it is required the online calculation of the image Jacobian, that is, a quantity depending on the distance between the target and the camera which is difficult to evaluate. A control in the image plane results also to be strongly nonlinear and coupled when mapped on the joint space of the robot and may cause problems when crossing points which are singular for the kinematics of the manipulator [1].

Visual servo systems can also be classified on the basis of their architecture in the following two categories [1]: the vision system provides an external input to the joint closed-loop control of the robot that stabilizes the mechanism (*dynamic look and move*); the vision system is directly used in the control loop to compute joints inputs, thus stabilizing autonomously the robot (*direct visual servo*). In general, most applications are of the dynamic look and move type; one of the reasons is the difference between vision systems and servo loops rates. The low frequency imposed by vision might cause problems on controller's stability, especially in cases where several DOFs are involved.

The aim of this work is to compare two different visual servoing controls, respectively, of the position-based and image-based type, implemented to govern a translating parallel manipulator with an eye-in-hand configuration. The robot must be able to reach and grasp a special target randomly positioned in the workspace; the control must be adaptive to compensate motions of the target in the 3D

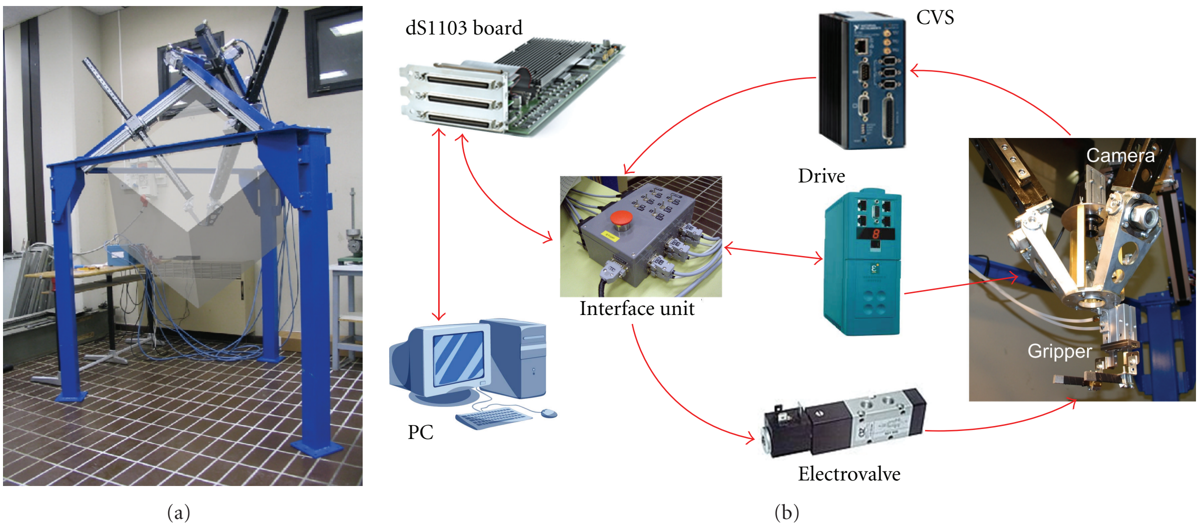


Figure 1: (a) ICaRo robot and workspace; (b) system architecture and connection diagram.

space. The use of Bézier curves [18, 19] in the trajectory planning algorithms ensures the continuity of the velocity vector, whereas a replanning event is needed. A dynamic look and move philosophy has been adopted to conjugate the low frame rate of the vision system (lower than 30 Hz) with the high control rate of the joint servo (about 1000 Hz).

2. Hardware Setup

The robot, which is called ICaRo, is a research prototype designed and realized at the Machine Mechanics Lab of Università Politecnica delle Marche [20]; the end-effector is in parallel actuated by 3 limbs whose kinematic structure only allows pure translations in the 3D space. The workspace is a cube of 0.6 m edge free of singular points (Figure 1(a)).

The eye-in-hand configuration has been chosen for the installation of the vision system. The optical axis of the camera is aligned with the end-effector vertical axis. The end-effector is also provided with a pneumatic gripper in order to grasp the target when the desired relative pose is reached (Figure 1(b)). The robot is managed by a central control unit, a DS1103 real-time board by dSPACE. The vision data, after image acquisition and preliminary processing made by the CVS real-time hardware of National Instruments, are sent via a serial interface to the central unit that runs the control algorithms.

3. PBVS Control

Using a position-based method, a 3D camera calibration is required in order to map the 2D data of the image features to the Cartesian space data. This is to say that intrinsic and extrinsic parameters of the camera must be evaluated. *Intrinsic parameters* depend exclusively on the optical characteristics, namely, lens and CCD sensor properties. The calibration of intrinsic parameters can be operated offline in the case that optical setup is fixed during the operative tasks of the robot. *Extrinsic parameters* indicate the relative pose of the camera reference system with respect to a generic world reference system. It is assumed that the world reference system is exactly the object frame, so that the extrinsic parameters give directly the pose of the camera with respect to the target. Obviously the extrinsic parameters are variable with robot or target motion, and an online estimation is needed in order to perform a dynamic look-and-move tracking task.

3.1. Estimation of Intrinsic Parameters. A standard pinhole camera model is here briefly described to introduce the notation used in this paper. A point in the 2D image space is denoted by $\mathbf{m} = [u \quad v]^T$, while a point in the 3D Cartesian space is denoted by $\mathbf{p} = [x \quad y \quad z]^T$. Introducing the homogeneous coordinates, it is $\tilde{\mathbf{m}} = [u \quad v \quad 1]^T$ and $\tilde{\mathbf{p}} = [x \quad y \quad z \quad 1]^T$. The image projection that relates a 3D point $\mathbf{p}$ to its corresponding point $\mathbf{m}$ in the image plane can be expressed as

$$s\tilde{\mathbf{m}} = \mathbf{A}\mathbf{\Pi}\mathbf{B}\tilde{\mathbf{p}} \qquad \mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{1}$$

where s is an arbitrary scale factor and $\mathbf{B}$ is the 4×4 matrix of the extrinsic parameters that combines a rotation and a translation relating the world coordinate system to the camera coordinate system. $\mathbf{A}$ is the matrix of intrinsic parameters, defined as

$$\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

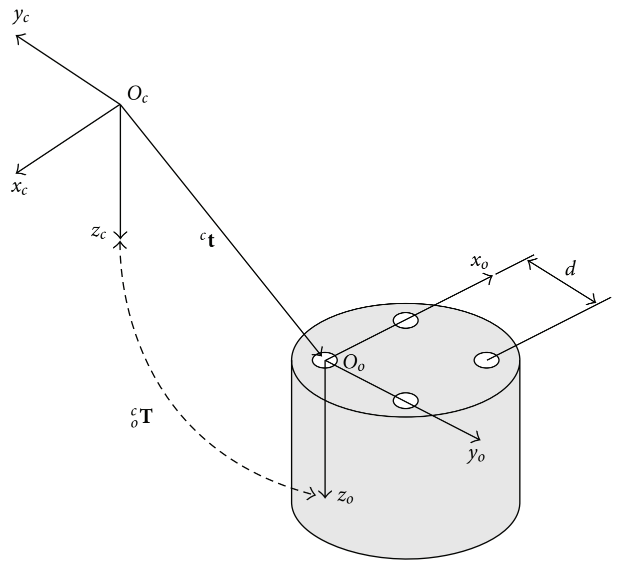


FIGURE 2: Object frame attached to the visual pattern and camera reference system.

where

(i) (u_0, v_0) are the coordinates of the principal point, that is, the projection of the optical axis on the CCD sensor;

(ii) α and β are the focal scale factors in the image u and v axes;

(iii) γ is the parameter describing the skewness of the two images axes, usually set to 0.

The pinhole model needs to be refined in order to correct image distortions introduced by the lens [21–25]. To this aim, normalized coordinates are conveniently introduced: given a point $\mathbf{p} = [x_c \quad y_c \quad z_c]^T$ in the camera reference system, the corresponding normalized pinhole projection is defined as $\mathbf{m}_n = [u_n \quad v_n]^T = [x_c/z_c \quad y_c/z_c]^T$. It can be demonstrated that, in the ideal case free of lens distortions, the intrinsic camera matrix $\mathbf{A}$ relates a point in the image plane with its normalized projection: $\tilde{\mathbf{m}}_n = \mathbf{A}^{-1}\tilde{\mathbf{m}}$. Actually, normalized coordinates obtained from previous equation are affected by image distortions, and the real case is distinguished from the ideal case by using the notation $\tilde{\mathbf{m}}_d = \mathbf{A}^{-1}\tilde{\mathbf{m}}$. Two components of distortion can be recognized: the radial lens distortion causes the actual image point to be displaced radially in the image plane; if centres of curvature of lens surfaces are not collinear, also a tangential distortion is introduced. The following expression is generally used to model both components of distortion:

$$\mathbf{m}_d = \mathbf{m}_n + \delta\mathbf{m}^{(r)} + \delta\mathbf{m}^{(t)}, \tag{3}$$

where $\delta\mathbf{m}^{(r)}$ and $\delta\mathbf{m}^{(t)}$ are defined as high-order polynomial functions of the undistorted coordinates u_n, v_n [21]. Because of the high degree of the distortion model, there exists no general algebraic expression for the inverse problem of computing the normalized image projection vector $\mathbf{m}_n$ from the distorted coordinates of vector $\mathbf{m}_d$. If low-distortion lenses are used, a direct solution can be found making the approximation of computing $\delta\mathbf{m}^{(r)}$ and $\delta\mathbf{m}^{(t)}$ as functions of distorted coordinates. Inspiring to well-known distortion models [21–25], a rough correction of the image distortion can be quickly obtained by using the equation

$$\mathbf{m}_n = \frac{1}{a}\left(\mathbf{m}_d - \begin{bmatrix} b & c \end{bmatrix}^T\right), \tag{4}$$

where

$$\begin{aligned} a &= 1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6, \\ b &= 2p_1 u_d v_d + p_2\left(r_d^2 + 2u_d^2\right), \\ c &= p_1\left(r_d^2 + 2v_d^2\right) + 2p_2 u_d v_d, \\ r_d^2 &= u_d^2 + v_d^2, \end{aligned} \tag{5}$$

and k_1, k_2, k_3, p_1, p_2 are additional intrinsic parameters to be defined by calibration.

Several algorithms for the estimation of intrinsic parameters are available in literature [22–25]. A technique inspired to the Heikkilä algorithm has been adopted: after the settings of the optical system have been tuned for an optimal result of the vision, a number $n > 4$ of frames (15 in our application) of a grid of known dimensions are acquired; through an automatic extraction of the corners of the grids, the algorithm is able to estimate with an iterative procedure the intrinsic parameters of the camera including distortion parameters, according to (4) and (5).

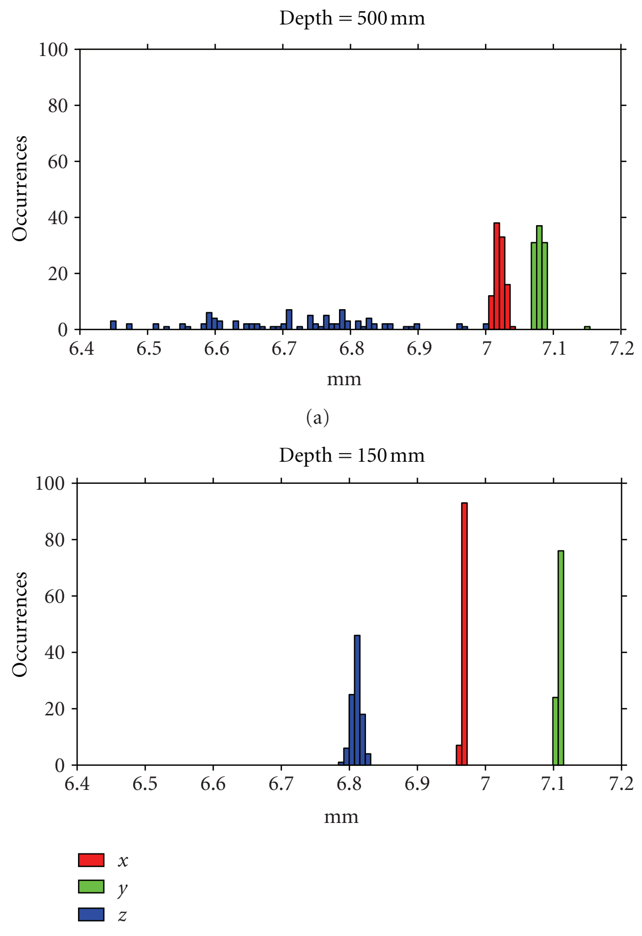
3.2. Estimation of Extrinsic Parameters. The reconstruction of the relative pose between camera and object is possible through the estimation of extrinsic parameters of the camera with respect to a reference system that is coincident with the object frame $O_o - x_o y_o z_o$. Figure 2 shows the optical pattern realized on the top surface of the target object; it consists of four coplanar circles positioned at the corners of a square of known edge d. The object frame is attached to the pattern in the way shown in the figure.

The position of the origin O_o with respect to the camera frame is the vector ${}^c\mathbf{t} = {}^c(O_o - O_c)$, while the relative orientation between the two reference systems is the rotation ${}^c_o\mathbf{R}$. The homogeneous transformation ${}^c_o\mathbf{T}$ coincides with the extrinsic parameters matrix

$$\mathbf{B} = {}^c_o\mathbf{T} = \begin{bmatrix} {}^c_o\mathbf{R} & {}^c\mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{6}$$

The aim is to determine the extrinsic parameters, once the geometry of the optical pattern and the pixel coordinates of the centroids of the circles projected onto the image plane are known.

The problem of determining the pose, knowing the correspondence of n points in the world and camera frame reference systems, is typical in photogrammetry (*PnP* problem), and it is proved that for 4 coplanar points the solution is unique [26, 27]. While the calibration of intrinsic parameters is quite laborious, but does not need to be



(a)

(b)

Figure 3: Distribution of displacement experimental data; (a) distance camera-target ≃500 mm; (b) distance camera-target ≃150 mm.

performed online, the solution of the *PnP* algorithm for 4 coplanar points is fast enough to be implemented online during the motion of the robot.

An experimental test is here presented in order to evaluate the accuracy of the implemented pose estimation method: while keeping fixed the camera, the optical target of Figure 2 is moved 7 mm along each axis using a precision micrometric stage. An image of the pattern is grabbed before the motion, and a series of 100 images are stored after the motion.

The relative displacements can be estimated by making the difference between the pose obtained from the first image and the pose obtained from each one of the further 100 images. The test has been repeated in two configurations, setting the initial z-distance between the camera and the object, respectively, at 500 mm and 150 mm.

Experimental data of the tests are plotted in Figure 3. It is clear that there is a larger dispersion of measurements at a bigger distance from the target; moreover, it can be noticed a larger sensitivity to noise along the z axis, parallel to the optical axis. It results also evident how the accuracy of the pose estimation increases approaching the target. The same conclusions can be drawn from Table 1, where experimental data are summarized in terms of mean values and standard deviations: passing from a distance of 500 mm to 150 mm, the standard deviation lowers nearly one order of magnitude in x and y and two orders in z.

Table 1: Summary of displacement experimental data: mean values and standard deviations of measured data.

Depth = 500 mm			
	x	y	z
Mean value [mm]	7.021	7.081	6.710
Standard deviation [mm]	0.007	0.009	0.134
Depth = 150 mm			
	x	y	z
Mean value [mm]	6.967	7.108	6.810
Standard deviation [mm]	0.001	0.001	0.007

3.3. *Control Algorithm.*

A dynamic look and move control has been implemented following the position-based approach above described. The global architecture is shown in Figure 4, where $\mathbf{s}$ represents the vector of image features, $\mathbf{q}$ is the vector of joint angles, and $\mathbf{x}_m, \mathbf{x}_d$, and $\mathbf{x}_{\text{plan}}$ are, respectively, the measured, desired, and planned Cartesian coordinates vectors. Looking at the scheme of the control, two separate loops can be identified: the inner loop realizes a standard PD control in the operative space with gravity compensation ($\mathbf{g}$ term) by exploiting the information from encoders; the camera, working as an exteroceptive sensor, closes an outer loop where visual information is processed in order to plan and replan in real time the desired trajectory of the robot. It is important to remark that image features extraction is performed externally from the control loop by the real-time image processing hardware (CVS); only the pose estimation, that includes the solution of the *PnP* algorithm, is involved in the loop.

A control in the Cartesian space is realized imposing the following requirements to the planned motions:

(a) starting from an arbitrary status of the end-effector (position and velocity), the continuity of position and velocity must be ensured;

(b) the final point must be reached with a vertical tangent in order to facilitate the grasping of the target object.

To this purpose, a third-order Bézier curve has been adopted [18, 19]. This curve is defined by 4 points in the 3D space and has the fundamental property of being tangent to the extremities of the polyline defined by the four points (Figure 5(a)).

The parametric formulation of a cubic Bézier curve is

$$\begin{aligned}\mathbf{B}(s) = \mathbf{P}_0\left(1-s^3\right) + 3\mathbf{P}_1 s(1-s)^2 + \\ 3\mathbf{P}_2 s^2(1-s) + \mathbf{P}_3 s^3 \quad s \in [0,1].\end{aligned} \tag{7}$$

In the trajectory planner algorithm, $\mathbf{P}_0$ is the current position of the end-effector, while $\mathbf{P}_3$ is the target point

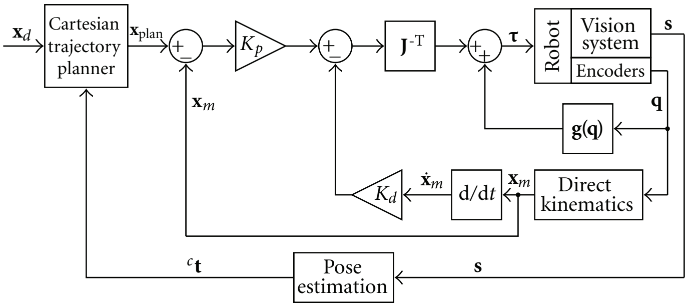
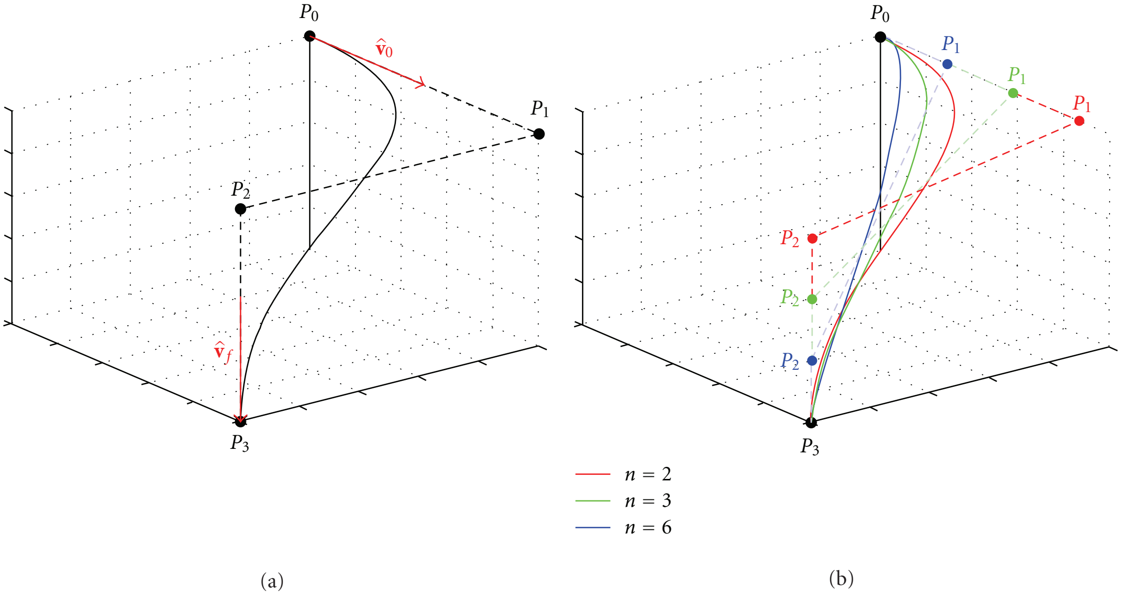


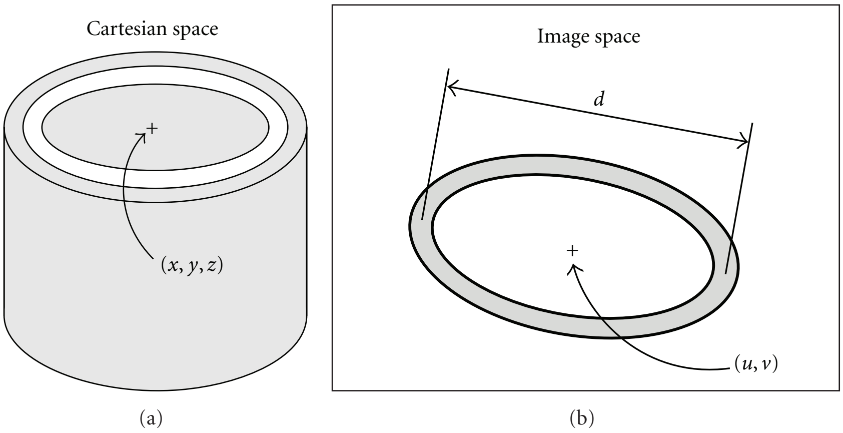
FIGURE 4: Dynamic look and move PBVS control.



(a)

(b)

FIGURE 5: (a) Plot of a cubic 3D Beziér curve; (b) influence of the parameter n on the planned trajectory.



(a)

(b)

FIGURE 6: Visual pattern for the IBVS: coordinates in the Cartesian and image spaces.

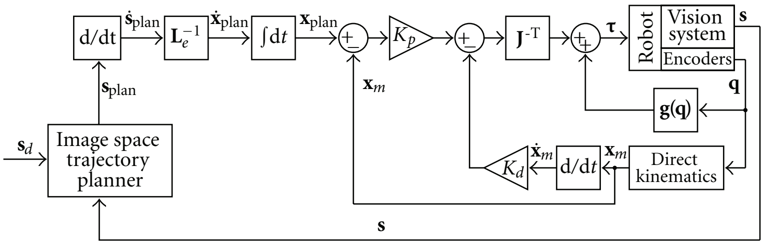
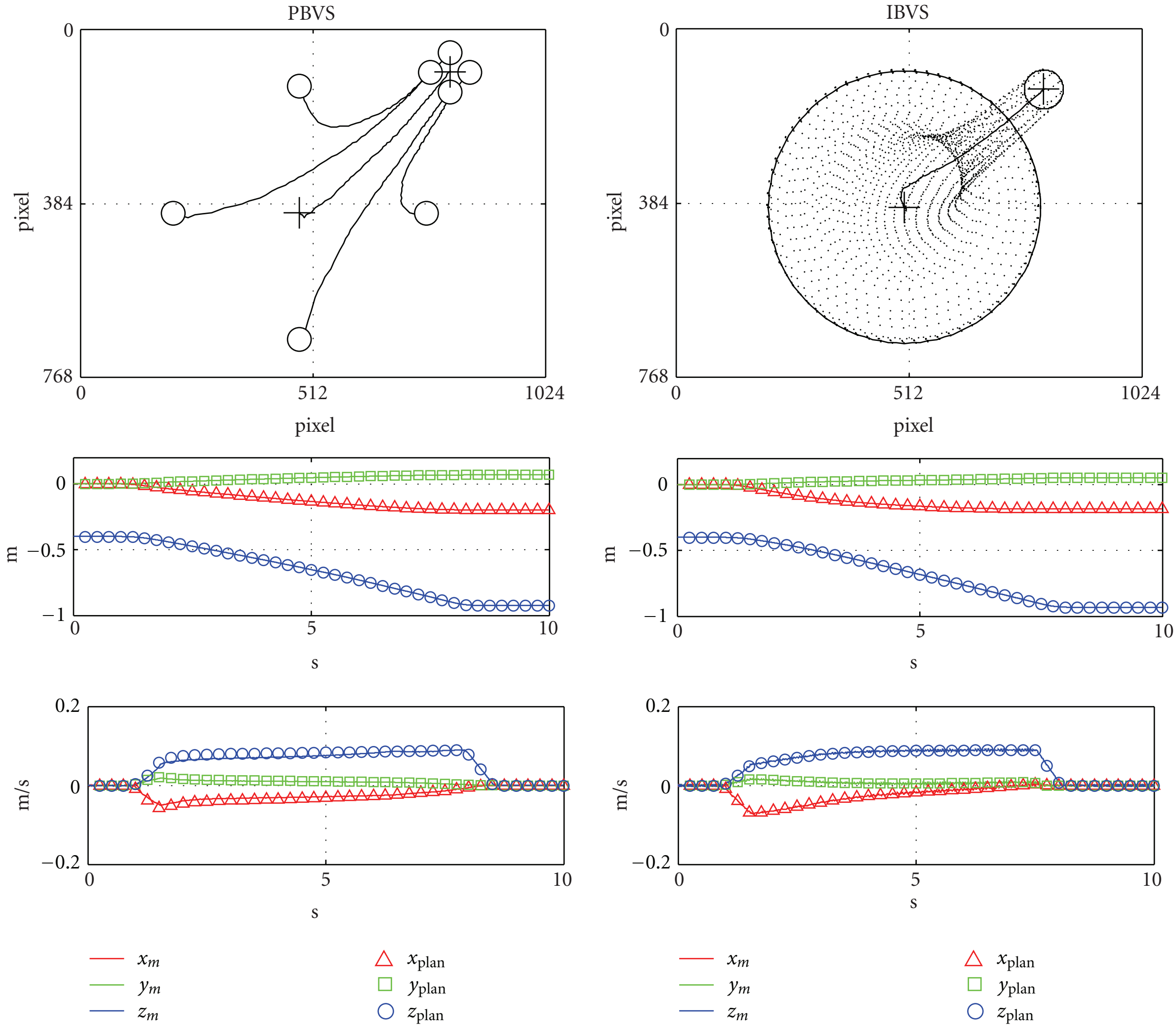


FIGURE 7: Dynamic look and move IBVS control.



(a)

(b)

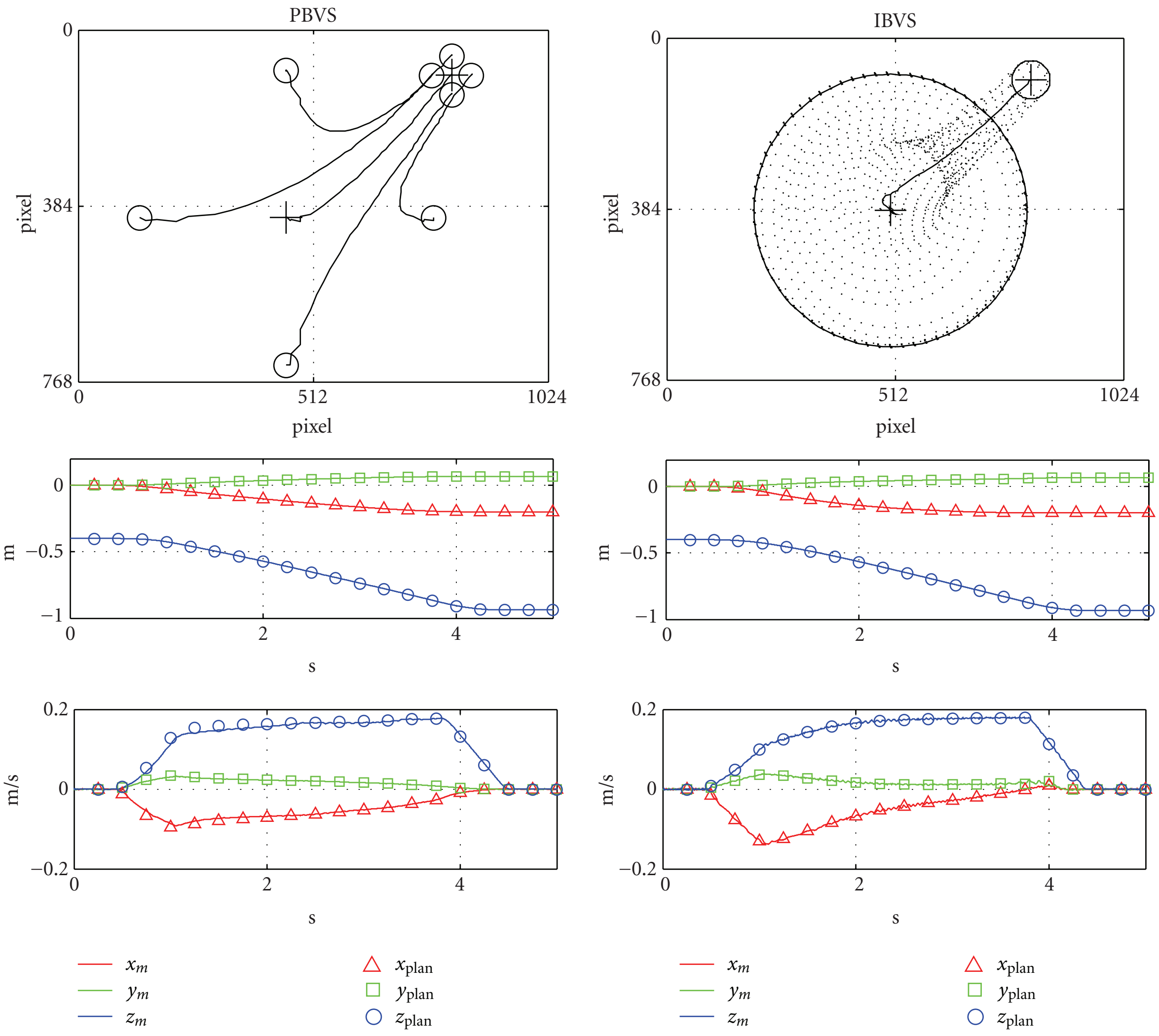
FIGURE 8: Slow tests ($v_{\max} = 10^{-1}$ m/s) with fixed target.

obtained from the vision system. Knowing the current velocity direction $\hat{\mathbf{v}}_0$ and the desired final velocity direction $\hat{\mathbf{v}}_f$, $\mathbf{P}_1$ and $\mathbf{P}_2$ are defined as

$$\mathbf{P}_1 = \mathbf{P}_0 + \frac{\|\mathbf{P}_3 - \mathbf{P}_0\|}{n}\hat{\mathbf{v}}_0,$$
$$\mathbf{P}_2 = \mathbf{P}_3 - \frac{\|\mathbf{P}_3 - \mathbf{P}_0\|}{n}\hat{\mathbf{v}}_f, \tag{8}$$

where n is a tunable parameter that influences the curvature of the trajectory (Figure 5(b)); all experimental data reported in the paper are obtained by setting $n = 3$.

Since the target $\mathbf{P}_3$ is continuously estimated during the motion of the robot, variations of its position due to motions of the object or to the reduction of the measurement errors (as shown in Figure 3) are compensated by a continuous replanning of the trajectory. A trapezoidal velocity profile



(a)

(b)

FIGURE 9: Fast tests ($v_{\max} = 2 \cdot 10^{-1}$ m/s) with fixed target.

is used with a temporal planning algorithm that is able to modify the shape of the profile, while the length of the planned trajectory changes during the motion.

4. IBVS Control

In the image-based visual servo control, the error signal is defined directly in terms of image feature parameters. In the present application, the optical target is an annulus painted on the top surface of a cylinder (Figure 6).

The image features are the coordinates of the center and the major axis of the ellipse which best approximates the projection of the annulus on the image plane; they are collected in the image space vector $\mathbf{s} = \{u, v, d\}^T$ expressed in pixel. The relation between image space and Cartesian space is given by in the interaction matrix $\mathbf{L_e}$ [27]: being $\dot{\mathbf{x}} = \{\dot{x}, \dot{y}, \dot{z}\}^T$ the Cartesian velocity and $\dot{\mathbf{s}} = \{\dot{u}, \dot{v}, \dot{d}\}^T$ the image space velocity, it is

$$\dot{\mathbf{s}} = \mathbf{L_e}\dot{\mathbf{x}}. \tag{9}$$

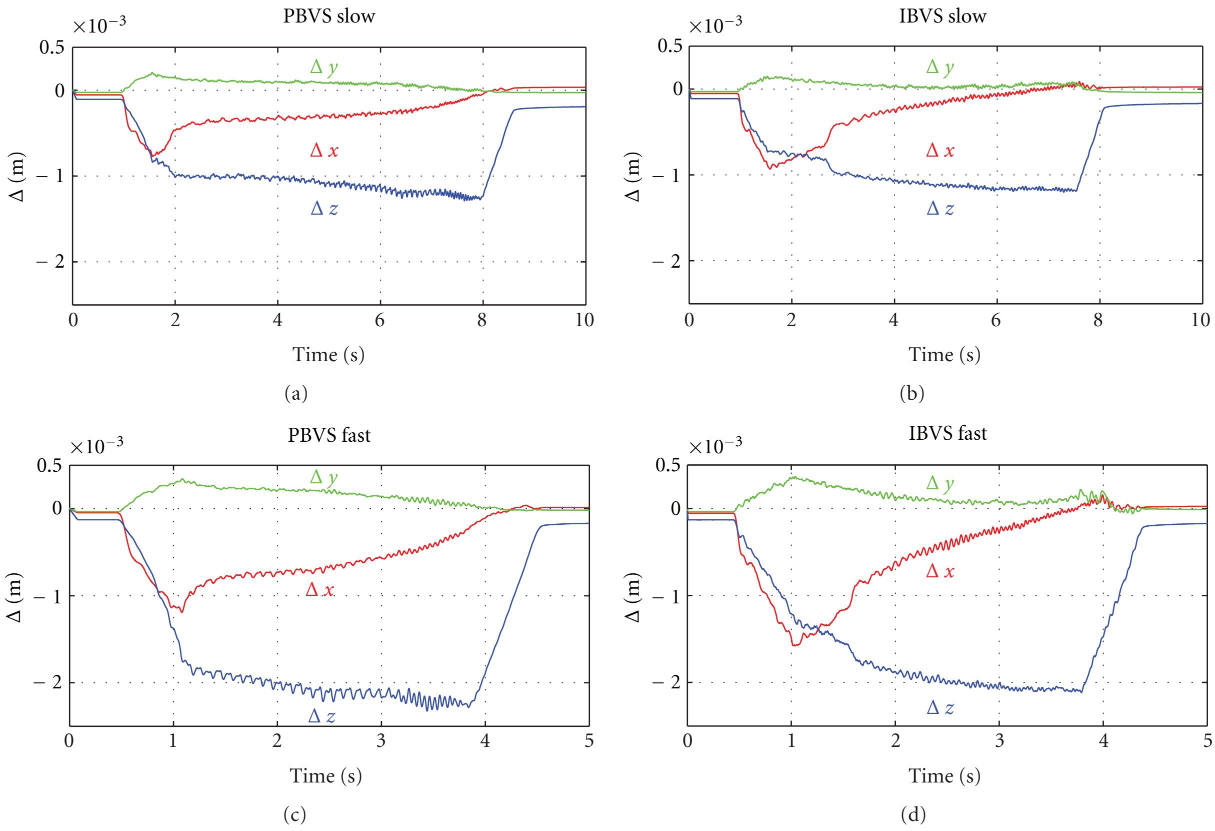
Here, the interaction matrix is defined as

$$\mathbf{L_e} = \begin{bmatrix} f/z & 0 & -u/z \\ 0 & f/z & -v/z \\ 0 & 0 & -fD/z^2 \end{bmatrix}, \tag{10}$$

where f is the focal length (assumed equal in u and v directions) expressed in pixel and D is the metric value of the annulus diameter in the Cartesian space. It arises from (10) that the interaction matrix is a function of image features u, v and of the Cartesian variable z. The estimation of z is performed by a linear interpolation on the parameter d; the interpolation law is obtained in a preliminary calibration step. The inversion of (9) is performed at each time step of the control, so that

$$\dot{\mathbf{x}} = \mathbf{L_e}^{-1}\dot{\mathbf{s}}. \tag{11}$$

The trajectory planner operates in this case over the image space: in analogy with the PBVS control, it is defined



(a)

(b)

(c)

(d)

Figure 10: Errors between planned and measured Cartesian coordinates for PBVS and IBVS at slow and fast mode.

as a Bézier curve according to (2) through points defined by image coordinates $\{u, v, d\}$. Such kind of strategy allows for the continuity of the velocity in both image and Cartesian spaces, even in the case of motion of the target.

Particular attention is focused on the definition of the velocity profile: without going into details for the sake of brevity, it can be summarized that velocity profiles on the image space are properly planned in order to obtain an effective trapezoidal profile in the Cartesian space. This operation is required to prevent that any component of the Cartesian velocity may assume peaks that are too high or may approach asymptotically zero during the motion of the robot. As an example, we may look to the relation that results from the classic pinhole camera model

$$d = \frac{fD}{z}. \tag{12}$$

The derivative of the above expression is:

$$\dot{z} = -\frac{\dot{d}}{fD}z^2, \tag{13}$$

which could be find directly from the term $\mathbf{L}_\mathbf{e}(3,3)$.

Thus, if a constant $\dot{d}$ is imposed according to the trapezoidal planning, the resulting Cartesian vertical speed $\dot{z}$ will approach zero when z approaches zero (i.e., when the camera approaches the target). Therefore, if the vertical speed has to be controlled, it must be directly planned in the Cartesian space, and analogously for the x and y components.

The scheme of the implemented IBVS, control is shown in Figure 7. As in PBVS it is possible to individuate the inner and outer loops. In the outer loop, the pose estimation block disappears, and the image features are directly the input of the trajectory planning block. Then the inverse of the interaction matrix is used to map the trajectory from image to Cartesian space according to (11).

5. Experimental Tests

A series of tests have been performed in order to compare the PB and IB approaches at different speeds and in cases of fixed or moving target. The starting position of the robot allows the camera to frame a wide area of the workspace; when the target is recognized by the vision system, the controller starts to govern the robot till the target is reached.

Figure 8 shows the results of the slow tests ($v_{\max} = 10^{-1}$ m/s) with a fixed target. The plots on the left side are referred to the PBVS, while IBVS results are on the right column; the position of the target is the same for both tests. The image plane trajectories of the features and the respective centroid are represented on the top of the figure; such features are the four coplanar circles for the PB and the annulus diameter (represented as a circle) for the IB. The Cartesian position and velocity are plotted, respectively, in

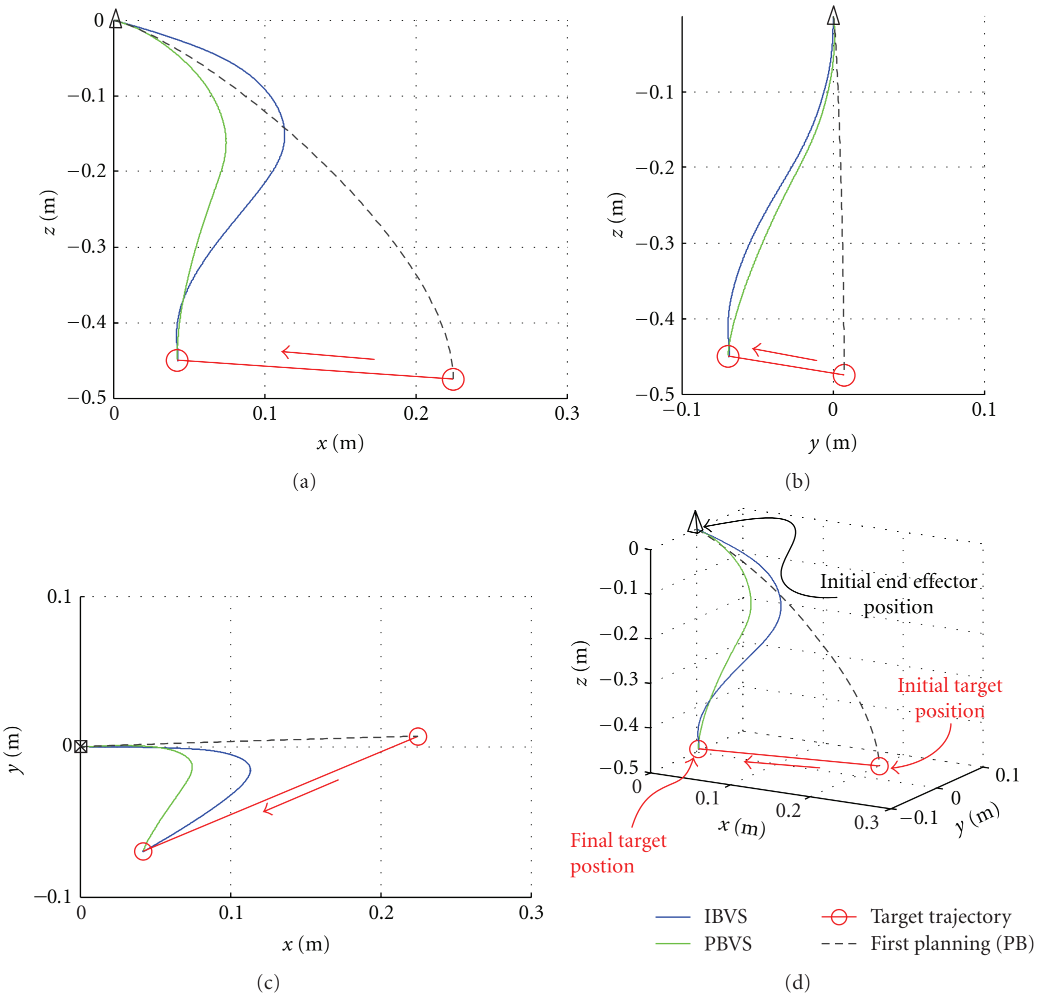


Figure 11: IBVS and PBVS tests with moving target.

the middle and at the bottom of the figure; lines refer to measured entities, while circular markers indicate planned values. Data show that the IB control has a slightly higher accuracy in centering the target. Further on, the IBVS presents higher x and y components of velocity in the initial part of the motion, which allows for quickly compensating the offset between camera and target in the horizontal plane.

Same considerations are even more evident in fast tests where the maximum velocity of the robot is twice that in the slow tests ($v_{\max} = 2 \cdot 10^{-1}$ m/s). Results are plotted in Figure 9 in analogy with Figure 8. Against a strong decrease of the time required to reach the target, it is not noticed for the IB control an appreciable loss of accuracy. For the PBVS, on the contrary, a certain increment of the final error is evident in the image plane.

For the sake of completeness, the error between planned and measured Cartesian coordinates in slow and fast tests is plotted in Figure 10. The comparison between IB and PB controls at the same speed shows that Cartesian errors are substantially similar. On the other hand, the error increases of about a factor 2 passing from the slow to the fast mode.

A further experimental evaluation is here described to prove the ability of the control in compensating dynamically a generic motion of the target: once the visual control has been started, the object is moved by means of a conveyor belt; the controller is able to replan continuously the trajectory ensuring the continuity of the velocity and quickly compensating the displacement imposed to the target. Tests are performed with $v_{\max} = 2 \cdot 10^{-1}$ m/s, which implies a time of about 5 s to reach the target. The original (referred to the PB algorithm) and replanned trajectories are plotted in Figure 11 in orthographic and axonometric views. The higher sensitivity to offsets on the horizontal plane is once again noticeable for the IBVS, which translates in higher performances in path following tasks.

6. Conclusions

Two dynamic look and move visual servos, respectively of the position-based and image-based type have been implemented to govern a parallel translating robot. Both controls are able to detect a target in the 3D workspace and to dynamically compensate a generic motion of the target itself. A continuous replanning algorithm defines the trajectory of the end-effector ensuring the continuity of the velocity vector even if sudden displacements are imposed to the target;

trajectories are planned using cubic Bézier curves that give a smooth spatial path and allow to impose the final direction of the velocity in order to facilitate the grasping task; such curves are planned, respectively, in the Cartesian space for the PBVS and in the image space for the IBVS.

Even if accuracies of the two proposed controls seem to be substantially comparable, experimental results suggest a better behaviour of the IBVS. Furthermore, the IBVS shows a greater sensitivity to horizontal displacement proved by higher components of the velocity vector in the x and y directions in the first part of the trajectory.

Both methods require the knowledge of the geometric model of the optical target; specifically for the proposed controls, the edge of the square formed by the four circles in the PB case and the diameter of the annulus in the IB case must be known in metric units. Nevertheless, the PB approach requires also the 3D calibration of the camera in order to estimate the intrinsic parameters matrix, which is then required for the execution of the *PnP* algorithm; as known from literature, the sensitivity to calibration errors is probably the main drawback of the PBVS.

Further on, even if the solution to the *PnP* problem returns the homogeneous transformation matrix ${}^{c}_{o}\mathbf{T}$, only the vector ${}^{c}\mathbf{t}$ is a useful information for a pure translation robot. It means that the PB approach gives redundant information for our specific purpose, which translates in useless computational effort for the real-time hardware.

All these reasons make the IBVS preferable for the studied application: the method is simple and accurate and does not require a camera calibration. Moreover, the risk of passing through or nearby singularities, which is typical for IB controls, is avoided if, as in this case, the robot is free of singular points inside its workspace.

Finally, it is worth to remark as the above considerations are strictly related to the kinematic properties of the robot and may not be valid a priori for other architectures. In this sense, authors are intended to implement PB and IB controls on a parallel robot which confers to the end-effector a 3DOFs motion of pure rotation. Such robot (SpheIRo) has been studied and prototyped at Machine Mechanics Lab of Università Politecnica delle Marche [28] with the philosophy of complementary mobility with respect to ICaRo. Unlike the translation manipulator, SpheIRo presents singular points inside its workspace which may cause problems of stability for the IB control. Moreover, passing from a relative position to a relative orientation problem, the sensitivity of the two algorithms in detecting the orientation between frames has to be assessed.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 651–670, 1996.

[2] P. I. Corke and S. A. Hutchinson, "Real-time vision, tracking and control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 622–629, April 2000.

[3] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[4] M. Staniak and C. Zieliński, "Structures of visual servos," *Robotics and Autonomous Systems*, 2010.

[5] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3D visual features," in *World Automation Congress, Robotics and Manufacturing Systems (WAC '96)*, vol. 3, pp. 497–502, 1996.

[6] W. J. Wilson, C. C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[7] N. R. Gans, A. P. Dani, and W. E. Dixon, "Visual servoing to an arbitrary pose with respect to an object given a single known length," in *Proceedings of the American Control Conference (ACC '08)*, pp. 1261–1267, June 2008.

[8] L. E. Weiss, A. C. Sanderson, and C. P. Neuman, "Dynamic visual servo control of robots: an adaptive image-based approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 662–668, 1985.

[9] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 5, pp. 691–700, 1989.

[10] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 2267–2271, April 1991.

[11] Z. Qi and J. E. McInroy, "Improved image based visual servoing with parallel robot," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 4, pp. 359–379, 2008.

[12] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.

[13] U. Khan, I. Jan, N. Iqbal, and J. Dai, "Uncalibrated eye-in-hand visual servoing: an LMI approach," *Industrial Robot*, vol. 38, no. 2, pp. 130–138, 2011.

[14] D. Fioravanti, B. Allotta, and A. Rindi, "Image based visual servoing for robot positioning tasks," *Meccanica*, vol. 43, no. 3, pp. 291–305, 2008.

[15] A. De Luca, G. Oriolo, and P. R. Giordano, "Image-based visual servoing schemes for nonholonomic mobile manipulators," *Robotica*, vol. 25, no. 2, pp. 131–145, 2007.

[16] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[17] N. R. Gans, S. A. Hutchinson, and P. I. Corke, "Performance tests for visual servo control systems with application to partitioned approaches to visual servo control," *International Journal of Robotics Research*, vol. 22, no. 10-11, pp. 955–981, 2003.

[18] T. W. Sederberg, "Computer aided geometric design," *BYU, Computer Aided Geometric Design Course Notes*, 2011.

[19] J. W. Choi, R. Curry, and G. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," in *Advances in Electrical and Electronics Engineering—IAENG Special Edition of the World Congress on Engineering and Computer Science (WCECS '08)*, pp. 158–166, October 2008.

[20] M. Callegari and M. C. Palpacelli, "Prototype design of a translating parallel robot," *Meccanica*, vol. 43, no. 2, pp. 133–151, 2008.

[21] D. C. Brown, "Decentering distortion of lenses," *Photometric Engineering*, vol. 32, no. 3, pp. 444–462, 1966.

[22] J. Heikkila and O. Silven, "Four-step camera calibration procedure with implicit image correction," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 1106–1112, June 1997.

[23] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[24] B. Bishop, S. Hutchinson, and M. Spong, "Camera modelling for visual servo control applications," *Mathematical and Computer Modelling*, vol. 24, no. 5-6, pp. 79–102, 1996.

[25] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[26] O. Faugeras, *Three Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Boston, Mass, USA, 1993.

[27] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, 2008.

[28] M. Callegari, "Design and prototyping of a spherical parallel machine based on 3-CPU kinematics," in *Parallel Manipulators: New Developments*, J.-H. Ryu, Ed., pp. 171–198, I-Tech, 2008.

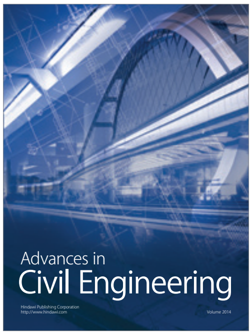
Journal of
Engineering

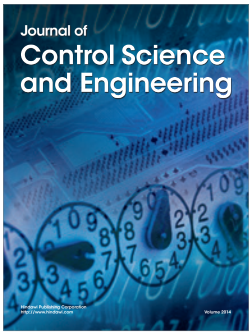
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

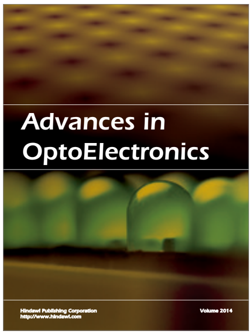
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

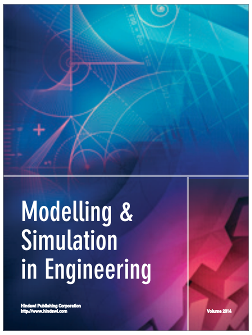
Journal of
Robotics

Journal of
Electrical and Computer Engineering

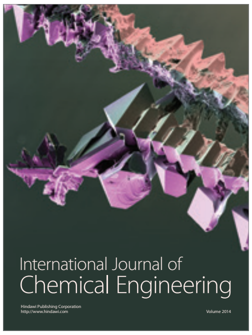

Advances in
OptoElectronics

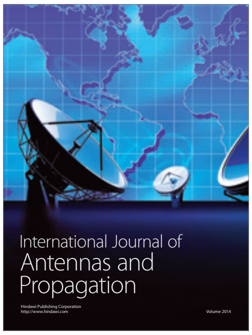
VLSI Design

International Journal of
Navigation and Observation

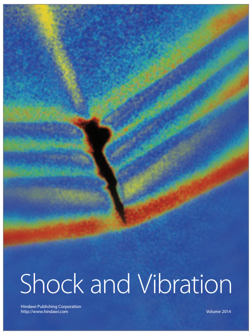

Modelling & Simulation in Engineering

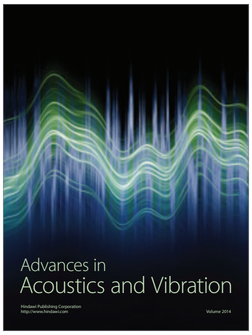
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration