Cartesian space visual control of a translating parallel manipulator

Massimo Callegari, Giacomo Palmieri, Matteo-Claudio Palpacelli

Department of Mechanics, Polytechnic University of Marche, Italy
E-mail: m.callegari@univpm.it, g.palmieri@univpm.it, m.palpacelli@univpm.it

Keywords: Visual Control, Parallel kinematics machines, Robotics.

SUMMARY. The paper presents the visual control of a parallel robot designed for motions of pure translation, called I.Ca.Ro. A camera is mounted directly on the robot in the eye-in-hand configuration; the vision data provide the information necessary for the following two control schemes: Position Based and Image Based Visual Servo. Such control architectures have been implemented on a dSPACE control board and the first experimental tests are currently being performed.

1 INTRODUCTION

Visual servoing is now one of the most interesting techniques for position control of manipulators because of the continuous development of hardware and software for image processing. By arranging a video-camera in an eye-in-hand configuration, or by using a set of cameras fixed to robot's frame, it is now possible to get quite fast feedback information about the pose of robot's end-effector; in this way the performances of the system and task accuracy can be improved by directly controlling the operational space variables.

A classification of visual based control systems can be made according to task space variables, namely *position-based* and *image-based* control [1,2]. Since both schemes present relevant features but drawbacks too, hybrid control architectures are often used even in industry to try to combine their advantages [3]: camera calibration errors generally reduce the robustness of position-based control applications, therefore image-based vision controls are mostly preferred to the formers. However *dynamic look-and-move* schemes are generally implemented in vision applications [4] in order to improve the internal stability of robots: these approaches are preferred to the *direct visual servo*, mentioned above, because vision data are available only at relative low sampling rate.

This paper presents the development of a position-based and an image-based control applied to a translating parallel manipulator characterized by a 3-CPU parallel kinematics [5]: the machine static and dynamic performances have been virtually investigated by means of the multibody software LMS Virtual.Lab and several control architectures have been implemented to assess the robot's closed-loop dynamic behaviour [6]. The integration of a vision system in the control architecture represents an attempt to overcome the problems connected to Cartesian positioning errors; in fact high bending moments by the joints located on the frame are caused by the heavy legs and moving platform and consequently the end-effector undergoes undesired displacements due to the elastic deformation of the structural elements. Moreover the application of a vision system can be exploited also for a robust and accurate calibration of the physical robot configuration so that the accuracy in positioning, typical of parallel robots, could be improved [7,8].

A first algorithm based on visual servoing was proposed in a previous work [9]: it is able to make the moving platform follow the path of a black disc that is manually moved over an horizontal plane lying in robot's workspace. A key issue in the algorithm is the evaluation of the extrinsic parameters of the camera, i.e. all the variables that connect the magnitudes measured on the camera reference system to robot’s global frame. The complexity of the algorithms described in [9] was relatively reduced because of the constancy of the camera extrinsic parameters, that were already known by camera calibration [10].

The present paper proposes an improvement of such algorithm that makes it able to handle displacements of the end-effector outside the previously defined planes, i.e. throughout all manipulator’s workspace: to this aim a more complicated routine has been implemented in order to evaluate on line the extrinsic parameters of the camera.

The two mentioned control architectures have been designed for a sample pick & place operation: the robot has to pick up a motionless target from a table moving itself from a starting pose to the fixed goal pose planned for the object's grip. Simple modifications can be made to the implemented routines in order to selectively pick up moving targets on a conveyor and place them in predefined areas.

2 DESCRIPTION OF THE SYSTEM

The present section briefly shows the system architecture in terms of interfaced devices, as shown in figure 1, and describes their functionalities. The 3-CPU robot, provided with the camera in the eye-in-hand configuration and a gripper, is managed by a central control unit, a DS1103 real-time board by *dSPACE*: the vision data processed by the Compact Vision System (CVS) standalone hardware of *National Instruments* are sent through a serial interface to a control unit that performs the vision control algorithms. All the connections are linked to a manufactured interface unit, where it is possible to manually enable drives and motors brakes. The goal is to grasp an object from a table by means of visual servoing techniques.

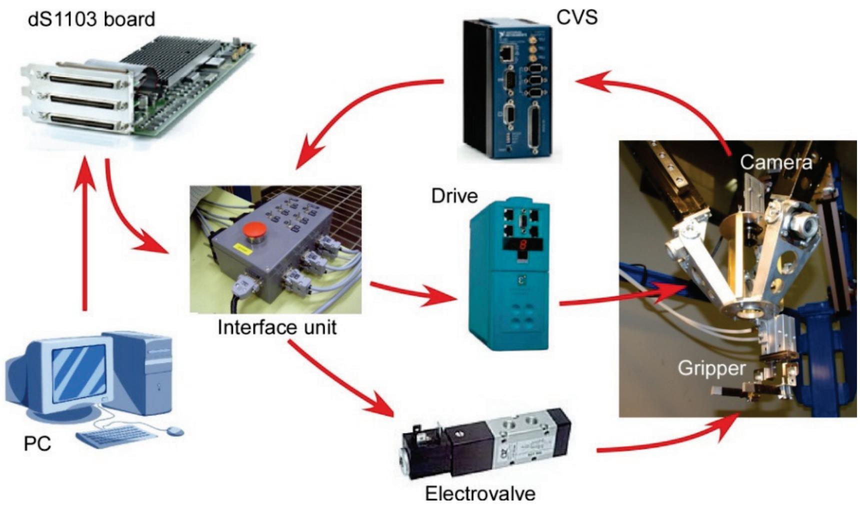
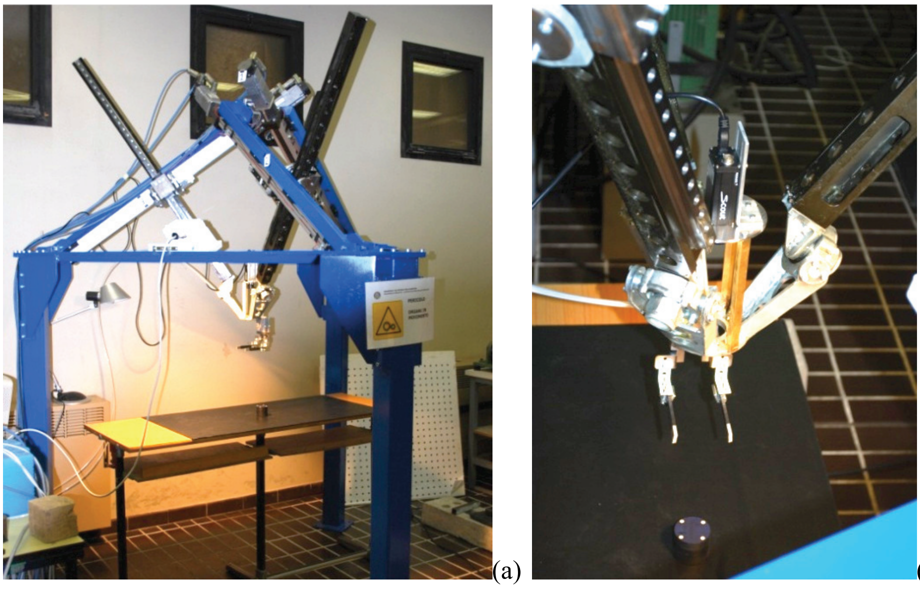


Figure 1: system architecture and connection diagram.

2.1 *I.Ca.Ro. parallel manipulator*

I.Ca.Ro., which stands for Innovative CArtesian Robot, is a prototype parallel kinematics machine that has been realised by the researchers of the Department of Mechanics at the Polytechnic University of Marche (see figure 2a): its mechanical architecture is based on the 3-CPU kinematics, i.e. the 3 legs connecting the mobile platform with the fixed base are characterised by cylindrical, prismatic and universal joints.

It has been demonstrated that the platform only translates in the space and that its Jacobian matrix represents a coordinates rotation between joint and cartesian space. Such matrix is always constant and positive, therefore yielding a workspace without singularities.



(a) (b)

Figure 2: the I.Ca.Ro. manipulator (a), vision and grasping rig (b).

2.2 *Vision and grasping rig*

A stand-alone device has been fully dedicated to image processing, in order to separate the vision tasks from the motion control (see figure 2b). The vision system has been programmed in such a way as to elaborate camera data at the maximum speed, then such information is sent to the central control hardware, that reads them when needed. A FireWire *Basler* high resolution CCD camera (ROI of 1024x768) with a 16 mm lens is fixed to the end-effector by means of a flange. This arrangement allows to keep the camera at an optimal distance from the table, ensuring a good depth of field and an acceptable acquisition area. The frame rate is set to the maximum (30 fps), in relation with the chosen camera resolution; exposure and focus are set manually before camera calibration and are kept fixed during platform's movements. The frame grabber is a real time embedded system by *National Instrument* (CVS-1456) that executes the vision processing algorithm with a FPGA logic. The CVS is programmable by the PC using *LabView* through an Ethernet connection and communicates with the *dSPACE* control unit by a serial port, sending strings of binary data.

In order to perform a pick up operation, a double effect servo-pneumatic gripper has been fixed to the end effector, screwed on a small flange in vertical position. Two long horizontal fingers are connected to the gripper to grasp the target when the gripper closes. All the components of the grasping system remain out of the field of view of the camera when the gripper is open during the visual served motion of the robot. When the desired relative position between the end-effector and the object is reached, the gripper is commanded to close. The control of the gripper is realised with a 3 way electro-valve supplied by a 7 bar pneumatic circuit, and the switch between "open" and "close" is governed by the *dSPACE* board that acts on the solenoid valve.

2.3 *Definition of the basic reference systems*

A coordinate system, defined as base system $O_q x_q y_q z_q$, can be associated to the fixed frame at the top of the robot, where the orthogonal cylindrical joints intersect at the common point O_q.: the three unit vectors are oriented along the slideways of the carriages of the linear modules. Similarly a coordinate system $O_e x_e y_e z_e$ can be conveniently introduced for the moving platform in a way that the z_e axis is directed vertically pointing downwards while the x_e and y_e unit vectors are located in the horizontal plane, as shown in figure 3; the origin O_e is located at the intersection of the inner revolute joints of the universal joints that connect the legs with the moving platform.

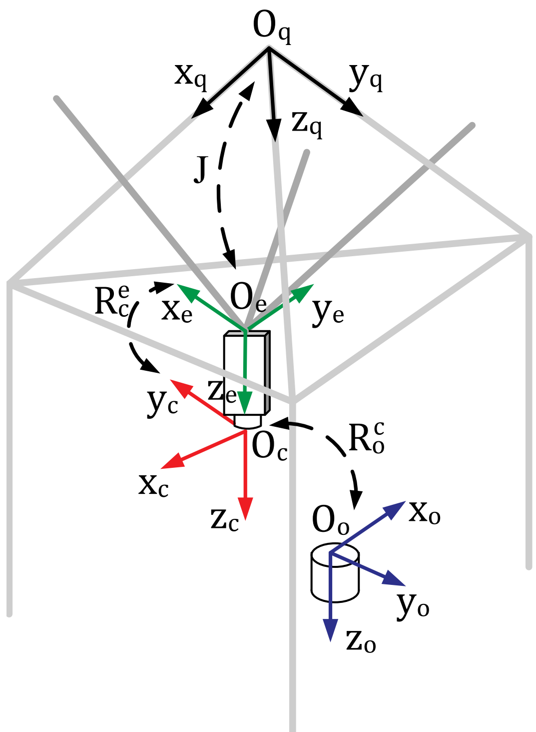


Figure 3: basic reference systems

It has been shown already that the I.Ca.Ro. kinematics is simple and characterised by a constant Jacobian matrix J: as a matter of fact, it is a rotation matrix mapping the velocity vector in the joint space, $\dot{\mathbf{q}}$ and the Cartesian velocity vector of the end-effector in the operational space, $\mathbf{v}$.

The camera is connected to the moving platform by means of a specially manufactured support that allows to move the robot avoiding collisions of the legs with the camera. A reference system $O_c x_c y_c z_c$ is attached at the camera at the lens centre: even if a desirable disposition of the camera is obtainable with an alignment of the z axes of the systems $O_e x_e y_e z_e$ and $O_c x_c y_c z_c$, such condition is not physically achievable because of the unavoidable geometrical errors of manufacture. Therefore a calibration procedure needed to work out the constant rotation matrix between them, $\mathbf{R}_e^c$, has been followed: an already implemented joint control has allowed to move the platform in the Cartesian directions x_e,y_e,z_e, observing in the meanwhile the readings in pixels of the camera. Eventually it has been possible to estimate directly each term $r_{i,j}$ in the i^{th} row and j^{th} column of the rotation matrix $\mathbf{R}_e^c$: for instance a displacement ${}^e\Delta z$ in the Cartesian space of the end-effector provides the three elements of the 3rd column of $\mathbf{R}_e^c$ by means of the following expressions

$$r_{1,3} = \frac{-\left({}^c\Delta\hat{x}\right)}{{}^e\Delta z} \qquad r_{2,3} = \frac{-\left({}^c\Delta\hat{y}\right)}{{}^e\Delta z} \qquad r_{3,3} = \frac{-\left({}^c\Delta\hat{z}\right)}{{}^e\Delta z} \tag{1}$$

where ${}^c\Delta\hat{x}$, ${}^c\Delta\hat{y}$, ${}^c\Delta\hat{z}$ are the estimated displacements in the camera system associated to the displacement vector (0, 0, ${}^e\Delta z$) defined in the end-effector reference system. The resulting matrix has been optimised by means of *Matlab Optimization Toolbox* to find an orthonormal transformation. It is interesting to note that the followed procedure has several limitations: first the Cartesian displacements ${}^e\Delta x$ and ${}^e\Delta y$ cannot be large because of the small camera acquisition area, so the results have a reduced accuracy, in addition it is necessary to know the actual robot kinematics, namely the real relationship between joint and Cartesian displacements.

Finally a coordinate system $O_o x_o y_o z_o$ is considered for the cylindrical target with the origin in the centroid of one of the white circles printed on its top surface and the axis z_o oriented like the cylinder axis. A further consideration can be made: since the cylindrical target is expected to be moved on an horizontal plane and the aim of the work is to pick it up from such plane, the last rotation $\mathbf{R}_o^c$ is not of interest in the present application, because it will be requested only the position of the centre of the target's top surface and the object is symmetrical respect to its vertical axis.

3 VISION ALGORITHMS

The aim of vision servoing for automated machines is the treatment of camera data to obtain information referred to operative space. The problem is to introduce a model to correlate 2D frame coordinates of a point into 3D world coordinates. The classical model for cameras is the pinhole, where camera parameters are divided in intrinsic and extrinsic. The first parameters are related to optical and hardware properties of the vision sensor: once estimated, they are constant if no adjustments are made to the optics. Intrinsic parameters are embedded in the 3x3 matrix $\mathbf{\Omega}$, see following section. Extrinsic parameters are related to the position and orientation of the camera with respect to a reference coordinate system; this is possible by defining a 4x4 matrix [**RT**], that is an homogeneous transformation of rotation and translation. They are the basic information to estimate the pose of the camera with reference to the object, or the opposite. The final relation between frame coordinates in pixels (u,v) and world coordinates (x,y,z) is expressed by:

$$\lambda \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{\Omega} \cdot \mathbf{\Pi} \cdot [\mathbf{RT}] \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \qquad \mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

where λ is an undefined scalar with $\lambda > 0$.

3.1 *Estimation of intrinsic parameters*

The matrix of intrinsic parameters is defined as:

$$\mathbf{\Omega} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where α and β are the focal lengths along the u and v directions of the CCD sensor, γ is the skewness (usually not relevant and set equal to 0), u_0 and v_0 are the projections of the optical axis on the CCD sensor. A calibration procedure is required to estimate these parameters and several algorithms are available in literature [11,12]. After the settings of the optical system have been tuned for an optimal result of the vision, a *Matlab* toolbox implementing the *Heikkikila* technique has been used to perform the calibration. The procedure consists on the acquisition on a number n>4 of frames (15 in our application) of a grid of known dimensions. After 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, and also the coefficients to correct the radial and tangential distortions of the lenses.

3.2 *Estimation of extrinsic parameters and of the pose of the camera*

Consider an eye-in-hand disposition of a camera with local reference system $O_cx_cy_cz_c$ and an object with coordinate system $O_ox_oy_oz_o$, as shown in figure 4. The position of the origin O_o with respect to the camera coordinate system is defined as **t**. The pose of the object with reference to the camera, that is the extrinsic parameters matrix [**RT**] in the pinhole model, is defined as:

$$\mathbf{T}_c^o = \begin{bmatrix} \mathbf{R}_c^o & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{4}$$

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 univocal [13]. It follows that, once intrinsic parameters have been evaluated, if the camera frames 4 coplanar points of a target, whose relative distances are known, it is possible to evaluate the position and orientation of the object in the camera coordinate system.

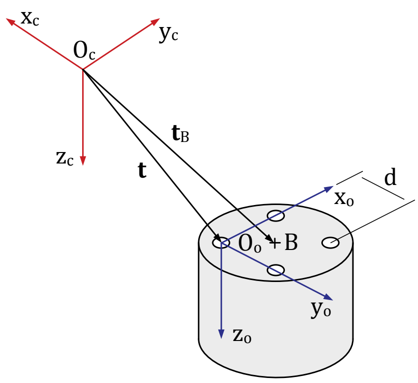


Figure 4: object and camera reference systems

While the calibration of intrinsic parameters is quite laborious, but does not need to be performed online, the solution of the P4P algorithm is fast enough to be implemented online during the motion of the robot. The image processing task is performed by the embedded vision system, that extracts the frame coordinates of the centroids of four white circles printed on the black surface of the target. The circles have a *5 mm* diameter and are positioned at the vertexes of a square with side of *30 mm*. Afterwards a blur filter reduces the noise of the image, a threshold makes the binarization between "dark" and "bright", extracting only the pixels relative to the circles. Then a particle analysis procedure calculates the coordinates of the centroids, which are sent by the serial port to the controller that solves the pose algorithm.

The vector **t** is referred to the origin of the object reference system, coincident with one of the four circles. Once the pose is determined, it is possible to obtain the position of the centre of the target respect to the camera:

$$\mathbf{t}_B = \mathbf{T}_c^o \cdot \begin{bmatrix} \frac{d}{2} & \frac{d}{2} & 0 \end{bmatrix}^T \tag{5}$$

where d is the distance between circles. Being the robot a translational machine, the orientation of the camera is fixed, while the orientation of the object is not known *a priori* and is generally variable during its motion. However the only significant information to insert in the control algorithm is the vector $\mathbf{t}_B$.

4 VISUAL CONTROL OF I.CA.RO.

A first attempt of introducing a vision system into a robotic architecture can be made by means of the two traditional visual servo control schemes, Position Based Visual Servo (PBVS) and Image Based Visual Servo (IBVS). They are respectively based on an interpretation of the features of the image, in conjunction with the knowledge of the geometric model of a target, and on the direct use of the image features. Their intrinsic nature entails some distinct characteristics: PBVS control needs a longer computational time and a routine to extract features from image, while IBVS presents the drawback of a non linear process highly coupled but it allows to eliminate errors in sensor modeling and camera calibration.

However their application to the I.Ca.Ro. translation platform proved to be easy because of the absence of rotational movements of the platform; moreover, as mentioned before, the coordinates transition between the reference systems of the camera and the end-effector is described by the constant rotation matrix obtained during the calibration phase.

4.1 *PBVS control scheme designed for the I.Ca.Ro. robot*

The control algorithm that has been implemented is a PD with gravity compensation in Cartesian space, exploiting the gravitational force vector already estimated in a previous work [14]. The functional scheme is shown in figure 5: the error ${}^c\mathbf{e}$ defined in the camera system is obtained by the difference between the desired position $\mathbf{t}_d$ and object's estimated position $\hat{\mathbf{t}}_B$. The vector $\hat{\mathbf{t}}_B$ results from the sequence of two operations, a P4P algorithm applied to the white circles centroids of the target and a FIR filtering to bring out the signal carrier and reduce the noisy. $\mathbf{K}_P$ and $\mathbf{K}_D$ have been chosen positive and diagonal in order to guarantee an asymptotic stability.

The complexity of the algorithm has been considerably reduced because of the constancy of the transformation between the reference systems defined in section 2.3.

The $\mathbf{t}_d$ vector represents the desired relative position between the object and the camera, expressed in the camera system, when the object is grasped by the gripper.

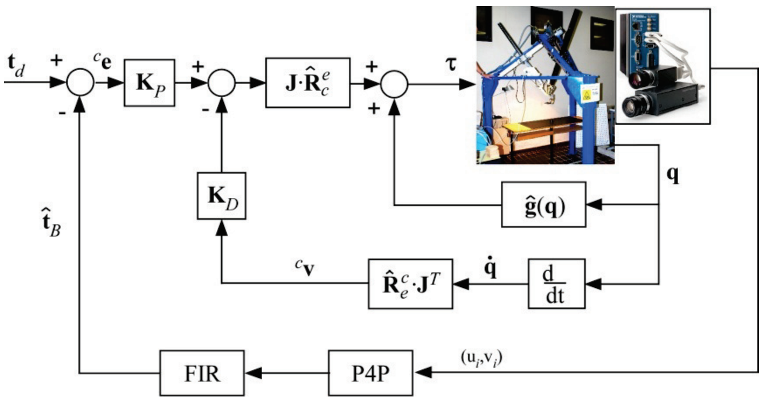


Figure 5: visual control PD with gravity compensation in Cartesian space

4.2 *IBVS control scheme designed for the I.Ca.Ro. robot*

Just as for the PBVS algorithm, a PD loop with gravity compensation has been used also for the IBVS case, but this time in the image plane. The vector $\mathbf{s_d}$ is given by the desired pose of the four centroids of the target's white circles and it is obtained from the reading of the camera when the robot is positioned in its goal position ready for grasping the object.

The error ${}^{S}\mathbf{e}$ is immediately available in the image data, in fact information in image plane about the position of the points in pixel can be easily converted in metric units by means of intrinsic parameters $\mathbf{\Omega}$. The vector $\mathbf{s}$ is given by $[x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4\,]^T$ where (x_i, y_i) is the 2D position in *mm* of the i^{th} centroid of the white circles.

An important tool required to develop an IBVS control is the *Interaction matrix* $\mathbf{L}_s$, that relates the time variation of image features $\dot{\mathbf{s}}$ to the velocity of the camera $\mathbf{v}_c$ when the target is fixed: in this case the velocity of the camera is the translational velocity of the robot's moving platform, which is rigidly connected to. The expression of the whole *Interaction matrix* $\mathbf{L}_s$ and of $\mathbf{L}_{s_i}$ for a single i^{th} point is here reported considering the nature of the robot [15]:

$$\mathbf{L}_s = \begin{bmatrix} \mathbf{L}_{s_1} \\ \mathbf{L}_{s_2} \\ \mathbf{L}_{s_3} \end{bmatrix} \qquad \text{with} \qquad \mathbf{L}_{s_i}(\mathbf{s}_i, Z_i) = \begin{bmatrix} -\dfrac{1}{Z_i} & 0 & \dfrac{x_i}{Z_i} \\ 0 & -\dfrac{1}{Z_i} & \dfrac{y_i}{Z_i} \end{bmatrix} \tag{6}$$

The Z_i coordinate is the depth of the centroid of the i^{th} white circle with respect to the camera: it can be calculated by means of the pose estimation previously defined but to avoid this heavy computation it can be considered constant using the data of the desired position.

In Image based application the matrix of interest is the pseudo-inverse of $\mathbf{L}_s$, namely

$$\hat{\mathbf{L}}_s^+ = \left(\hat{\mathbf{L}}_s^T \hat{\mathbf{L}}_s\right)^{-1} \hat{\mathbf{L}}_s^T \tag{7}$$

whose dimension in this case is 3x8 and it is of rank 3. The small cap used refers to estimated magnitudes because the Z_i values have to be calculated at each iteration of the control scheme, but in this case it is used the constant matrix $\mathbf{L}_s^+(\mathbf{s}_d,\mathbf{Z}_d)$. The gravity compensation reflects the PBVS algorithm together with the matrix $\mathbf{K}_{Ps}$ and $\mathbf{K}_{Ds}$ that this time operate in image plane variables.

The stability of the control system is achieved by the implemented control law that provides the actuation torques to the motors:

$$\boldsymbol{\tau} = \hat{\mathbf{g}}(\mathbf{q}) + \left(\mathbf{J}\hat{\mathbf{R}}_c^e\hat{\mathbf{L}}_s^T\right)\cdot\left(\mathbf{K}_{Ps}\,{}^s\mathbf{e} - \mathbf{K}_{Ds}\hat{\mathbf{L}}_s\hat{\mathbf{R}}_e^c\mathbf{J}^T\dot{\mathbf{q}}\right) \tag{8}$$

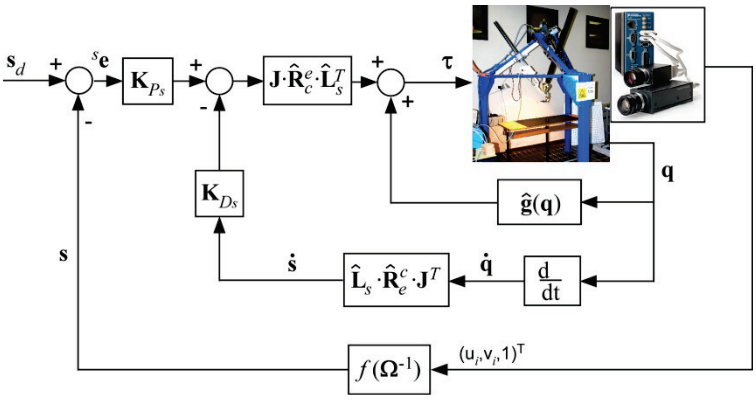


Figure 6: visual control PD with gravity compensation in Image plane

5 CONCLUDING REMARKS

A Position-based and an Image-based visual servo controls have been developed for a translating parallel kinematics machine: a black cylinder with four white circles painted on its top planar surface has been used as target for the visual controller. A calibration procedure of the camera has been implemented to obtain the camera intrinsic parameters: such information has been used to estimate the pose of the object. A CVS hardware elaborates images and sends high-level data to the control unit by a serial interface, therefore unloading the controller from heavy image processing.

The implementation of such algorithms on a dSPACE board have shown some important key points in the preliminary phases of controller design: the information that comes from the camera is sent at a slow rate and is quite variable, therefore the resolution of the camera has to be optimised in order to increase the frame rate but without reducing excessively the acquisition area, and a signal filtering is necessary to reduce the reading noise. Another important aspect, brought out from tests, is the estimation of the rotation matrix between the camera and the end-effector. Referring to PBVS, the controller directly operates on the camera positioning error producing a force vector defined in the camera system. Such vector can be expressed in the end-effector system by means of the mentioned matrix, therefore, if there is a significant error in its definition, the controller actually produces a wrong force vector in the end-effector system. It follows that the displacement of the end-effector will not be in the required direction and the displacement error in the camera system will not tend to the null value.

Presently the robot is able to reach the object through the proposed routines and grasp it but some refinements are necessary to obtain smooth movements of the moving platform; during the early tests, IBVS control has visibly shown a better behaviour, while the PBVS control has often lost the object's frame.

References

[1] Corke, P., *Visual Control of Robots: High-Performance Visual Servoing*, Mechatronics, Research Studies Press (John Wiley), (1996).

[2] Fioravanti, D., Allotta, B. and Rindi, A., "Image based visual servoing for robot positioning tasks," *Meccanica*, **43**(3), 291-305 (2008).

[3] Xu, D., Wang, L. and Tan, M., "Image Processing and Visual Control Method for Arc Welding Robot," in *Proc. IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, August 22-26, 2004.

[4] Lippiello, V., Siciliano, B. and Villani, L., "Position-Based Visual Servoing in Industrial Multirobot Cells Using a Hybrid Camera Configuration," *IEEE Transactions on Robotics*, **23**(1), 73-86 (2007).

[5] Callegari, M. and Palpacelli, M.-C., "Prototype design of a translating parallel robot," *Meccanica*, **43**(2), 133-151 (2008).

[6] Callegari, M., Palpacelli, M. and Ricci, S., "Controller design of the innovative I.Ca.Ro. parallel robot," in *Proc. RAAD International Workshop on Robotics in Alpe-Adria-Danube Region*, Ljubljana, Slovenia, June 7-9, 2007.

[7] Daney, D., Andreff, N., Chabert, G. and Papegay, Y., "Interval method for calibration of parallel robots: vision-based experiments," *Mechanism and Machine Theory*, **41**, 929-944 (2006).

[8] Meng, Y. and Zhuang, H., "Autonomous robot calibration using vision technology," *Robotics and Computer-Integrated Manufacturing*, **23**, 436-446 (2007).

[9] Palpacelli, M. and Palmieri, G., "Visual Based Control System of a 3-CPU Translational Parallel Manipulator," in *Proc. RAAD International Workshop on Robotics in Alpe-Adria-Danube Region*, Ancona, Italy, September 15-17, 2008.

[10] Traslosheros, A., Sebastián, J.M., Ángel, L., Roberti, F., Carelli, R., "Visual Servoing of a Parallel Robot System," *WISP 2007, IEEE International Symposium on Intelligent Signal Processing* , October 3-5, 2007.

[11] Heikkilä, J. and Silvén., O., "A Four-step Camera Calibration Procedure with Implicit Image Correction," In *Proc. CVPR Computer Vision and Pattern Recognition, IEEE Computer Society*, pp.1106-1112 (1997).

[12] Zhang, Z. "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(7), pp.1330–1334 (2000).

[13] Faugeras, O., *Three Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, Boston, MA, (1993).

[14] Palpacelli, M.-C., "Design and Prototyping of an Innovative Cartesian Parallel Manipulator," *Ph.D. dissertation,* Univ. Polytechnic of Marche, Ancona, Italy, (2007).

[15] Siciliano, B., Khatib, O. (Eds.), *Springer Handbook of Robotics*, (2008).