

Omnidirectional Visual-Servo of a Gough-Stewart Platform

Omar Tahri, Youcef Mezouar, Nicolas Andreff and Philippe Martinet

Abstract—This work deals with the control by vision of the Gough-Stewart platform. For that, a central catadioptric camera is used to observe the platform legs. This allows to obtain a large field of view, and then avoids the occlusion problems observed when a classical perspective camera is used. The leg projections onto the catadioptric plane are used to determine their orientation in the camera frame. Finally, the computed orientations will be used in a visual servoing scheme of the platform effector.

I. INTRODUCTION

Recently, it has been shown that visual servoing is a good way to control parallel mechanisms [1], [2], [12]. In [12], the end effector pose is measured by vision and used for regulation. However, the direct application of visual servoing techniques assumes implicitly that the robot inverse differential kinematic model is given and that it is calibrated. In [1], [2] propose respectively an image-based and position-based visual-servo schemes by observing the platform legs with classical perspective camera. In [1], the authors used the computed legs orientation from their image to control the end effector position. In [2], it is shown that better results could be obtained using the legs projection in image for regulation without any 3D reconstruction.

Unfortunately, to position adequately the camera to observe simultaneously all the platform legs is a complex task. In [1], [2], the camera was positioned in front of the platform (see Fig 1.a). In this case, the legs in the front of the platform are closer to the camera than the ones in the back. As a consequence, the extraction of the image features lying on the front legs will be more robust. Furthermore, large parts of the legs in the back can be occluded by the front legs (see Fig 1.b). This is an important drawback since the vision based control assumes that all legs can be observed during the servoing task. A first solution to address this issues is to employ a system made of multiple cameras. However, in this case, data provided by each camera must be synchronized and calibrated. A second and simpler solution consists on positioning a single omnidirectional camera at the platform center (see Figure 2.a). In such a way, all the legs can be simultaneously observed on panoramic view and potential occlusions can not occur (see Figure 2.b). An omnidirectional camera should highly increase the robustness of visual control of the Gough-Stewart platform. We will demonstrate this point in this paper.

Omnidirectional cameras are usually intended as a vision system providing a 360° panoramic views of the scene.

The authors are with LASMEA-CNRS, Université Blaise Pascal/IFMA, 63175 Aubière, France. They would thank Tej Dallej for substantial help during the realization of this work.

firstname.lastname@lasmea.univ-bpclermont.fr



Fig. 1. A Gough-Stewart platform observed by a classical perspective camera: (a) camera position with respect to the platform legs, (b) leg images

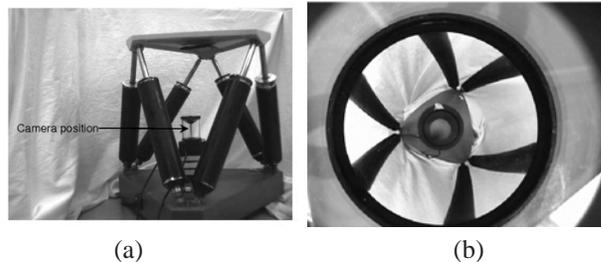


Fig. 2. A Gough-Stewart platform observed by an omnidirectional camera: (a) camera position with respect to the platform legs, (b) leg images

Such an enhanced field of view can be achieved by either using catadioptric systems, obtained by opportunely combining mirrors and conventional cameras, or employing purely dioptric fish-eye lenses [3]. In practice, it is highly desirable that such imaging systems have a single viewpoint [3], [11]. That is, there exists a single center of projection, so that, every pixel in the sensed images measures the irradiance of the light passing through the same viewpoint in one particular direction. The reason why a single viewpoint is so desirable is that it permits the generation of geometrically correct perspective images from the pictures captured by the omnidirectional camera. Geyer [6] and Barreto [5] developed a unified projection model for these sensors using properties of the projection of points on a sphere. In this work, this unified model will be used, since it allows to formulate control laws that are valid for any sensor in this class: perspective, catadioptric or fish-eye.

To date, as far as we know there is no work coupling the use of central catadioptric camera and parallel robots control. As we can see on Figure 2.b, the omnidirectional camera allows us to observe all legs without any occlusion. Moreover, by putting the camera in the middle of the legs, the latter are more closer to the image plane compared to in the classical camera case. The features extraction might thus be more robust. From the same Figure, we also note that

the leg positions in the image are almost radials. We will take advantage of this configuration to develop an automatic detection of them in image (see paragraph IV-A).

In the next section, we remind the unified catadioptric camera model. In section 3, the cylindrical leg observation and the interaction matrix related to the legs orientation in the camera frame are recalled. In Section 4, an automatic legs detection in image is proposed. The leg orientations in the camera frame are computed from their image, and used to control the platform effector position.

II. CENTRAL CATADIOPTRIC CAMERA MODEL

For sake of completeness, we present here a slightly modified version of the projection model of Geyer [6] and Barreto [5] (Fig. 3). The projection of 3D points can be done in the following steps:

- 1) world points in the mirror frame are projected onto the unit sphere,

$$(\mathcal{X}_{\mathcal{F}_m}) \rightarrow (\mathcal{X}_s)_{\mathcal{F}_m} = \frac{\mathcal{X}}{\|\mathcal{X}\|} = (X_s, Y_s, Z_s) \quad (1)$$

- 2) the points coordinates are then changed to a new reference frame centered in $\mathbf{p} = (0, 0, \xi)$,

$$(\mathcal{X}_s)_{\mathcal{F}_m} \rightarrow (\mathcal{X}_s)_{\mathcal{F}_p} = (X_s, Y_s, Z_s - \xi) \quad (2)$$

- 3) we then project the point onto the normalized plane,

$$\mathbf{m} = (x, y, 1) = \left(\frac{X_s}{Z_s - \xi}, \frac{Y_s}{Z_s - \xi}, 1 \right) = \tilde{h}(\mathcal{X}_s) \quad (3)$$

- 4) the final projection involves a generalized camera projection matrix \mathbf{K} (with f the focal length, (u_0, v_0) the principal point, s the skew and r the aspect ratio)

$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} \gamma & \gamma s & u_0 \\ 0 & \gamma r & v_0 \\ 0 & 0 & 1 \end{bmatrix} = k(\mathbf{m}) \quad (4)$$

The function \tilde{h} is bijective and

$$\tilde{h}^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{-\xi - \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} x \\ \frac{-\xi - \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} y \\ \frac{-\xi - \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} + \xi \end{bmatrix} \quad (5)$$

In this paper, the whole calibration parameters are supposed to be known [10]. In the next section we remind an adequate model of the platform legs in camera frame.

III. MODELING

A. Cylindrical leg observation

The legs are supposed to be cylindrical. Therefore, their projections are defined by the so-called interpretation planes. Define the normal unitary vector $\mathbf{n} = [n_x \ n_y \ n_z]^T$ as the orthogonal to the interpretation plane Π defined by a line in 3D space and the principal projection center. Thus the

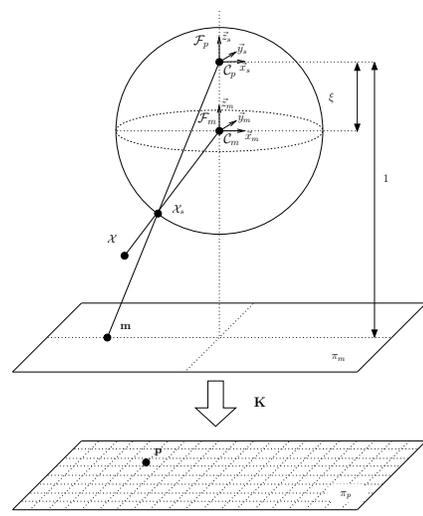


Fig. 3. Unified projection model

coordinates of the points belonging to the interpretation plane hold the following equation:

$$\mathbf{X} = [X \ Y \ Z]^T \in \Pi \iff n_x X + n_y Y + n_z Z = 0 \quad (6)$$

Let \mathcal{S} be the intersection between the interpretation plane and the sphere. By combining (1) and (6), \mathcal{S} is then defined by:

$$\begin{cases} X_s^2 + Y_s^2 + Z_s^2 = 1 \\ n_x X_s + n_y Y_s + n_z Z_s = 0 \end{cases} \quad (7)$$

Using the spherical coordinates given by (5), a line in space is thus mapped onto the image plane to a conic curve, which can be written as:

$$\alpha_0 x^2 + \alpha_1 y^2 + 2\alpha_2 xy + 2\alpha_3 x + 2\alpha_4 y + \alpha_5 = 0 \quad (8)$$

with:

$$\begin{cases} \alpha_0 = n_x^2 - \xi^2(1 - n_y^2) \\ \alpha_1 = n_y^2 - \xi^2(1 - n_x^2) \\ \alpha_2 = n_x n_y (1 - \xi^2) \\ \alpha_3 = n_x n_z \\ \alpha_4 = n_y n_z \\ \alpha_5 = n_z^2 \end{cases} \quad (9)$$

The coefficient of (8) are defined with a scale factor. If $\alpha_5 \neq 0$, the number of those equation parameters can be reduced and (8) can be written as:

$$\beta_0 x^2 + \beta_1 y^2 + 2\beta_2 xy + 2\beta_3 x + 2\beta_4 y + 1 = 0 \quad (10)$$

with $\beta_i = \frac{\alpha_i}{\alpha_5}$. From the parameters β_i , it is possible to determine the perpendicular vector to the interpretation plane as follow:

$$\begin{cases} n_z = (\beta_3^2 + \beta_4^2 + 1)^{-\frac{1}{2}} \\ n_x = \beta_3 n_z \\ n_y = \beta_4 n_z \end{cases} \quad (11)$$

A forward determination of the perpendicular vector to the interpretation can also be obtained from α_i by:

$$\underline{\mathbf{n}} = \frac{\begin{pmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{pmatrix}}{\left\| \begin{pmatrix} \alpha_3 \\ \alpha_4 \\ \alpha_5 \end{pmatrix} \right\|} \quad (12)$$

The case where $\alpha_5 = 0$ corresponds to a degenerate case where the z -axis belongs to the interpretation plane. In this case, the projection of the leg edges is straight lines. Unfortunately this happens for several end-effector pose. Therefore, the estimation of $\underline{\mathbf{n}}$ using (11) or (12) from α_i or β_i will not be suitable, since $\alpha_3 = \alpha_4 = \alpha_5 = 0$. For this reason, a more robust estimation of $\underline{\mathbf{n}}$ from the projection onto a sphere projection will be proposed in the following of this paper. The proposed method is based on simple linear minimization. Further, the vector $\underline{\mathbf{n}}$ can be computed whatever the leg position (i.e. no degenerate case). More details about line projection with catadioptric cameras are given on [7]. Finally, the leg orientation, expressed in the camera frame, can straightforwardly computed as follow:

$${}^c \underline{\mathbf{u}}_i = \frac{{}^c \underline{\mathbf{n}}_i^1 \times {}^c \underline{\mathbf{n}}_i^2}{\| {}^c \underline{\mathbf{n}}_i^1 \times {}^c \underline{\mathbf{n}}_i^2 \|} \quad (13)$$

The measure ${}^c \underline{\mathbf{u}}_i$ will be used in this paper to servo the end-effector pose of the robot. In fact, by nature, this feature does not depend on the kind of the camera since the latter holds the unified model recalled above: classical perspective, central catadioptric or fish-eye. Thus the results presented in this paper are general, excepted the leg edges detection which depend on the camera kind. Otherwise, the features determination, the interaction matrix formulas and the control law are still the same in both cases. This was expected since the used features depend only on the frame where it is defined.

B. Robot modeling

The length of each robot leg holds the vector form, introduced as the *vision-based kinematics of the hexapod* expressed in the camera frame in [2]:

$$q_i {}^c \underline{\mathbf{u}}_i = {}^c \mathbf{R}_e {}^e \mathbf{B}_i + {}^c \mathbf{t}_e - {}^c \mathbf{A}_i \quad (14)$$

where ${}^c \mathbf{R}_e$ and ${}^c \mathbf{t}_e$ are respectively the orientation and position of the end-effector frame with respect to the base frame and where the left superscript denotes the reference frame in which the coordinates are taken, ${}^e \mathbf{B}_i$ are the attachment points of the legs to the moving platform (end-effector) by spherical joints, ${}^c \mathbf{A}_i$ are the attachment points of the legs to the base. The point coordinates ${}^c \mathbf{A}_i$ and ${}^e \mathbf{B}_i$ are hence constants. It follows that the derivative of (14) with respect to time is thus given by:

$$\dot{q}_i {}^c \underline{\mathbf{u}}_i + q_i \dot{{}^c \underline{\mathbf{u}}_i} = {}^c \dot{\mathbf{R}}_e {}^e \mathbf{B}_i + \dot{{}^c \mathbf{t}}_e \quad (15)$$

From (14), we have:

$${}^e \mathbf{B}_i = {}^c \mathbf{R}_e^T ({}^c \mathbf{t}_e - {}^c \mathbf{A}_i - q_i {}^c \underline{\mathbf{u}}_i) \quad (16)$$

By combining (15) and (16), we obtain:

$$\dot{q}_i {}^c \underline{\mathbf{u}}_i + q_i \dot{{}^c \underline{\mathbf{u}}_i} = {}^c \dot{\mathbf{R}}_e {}^c \mathbf{R}_e^T ({}^c \mathbf{t}_e - {}^c \mathbf{A}_i - q_i {}^c \underline{\mathbf{u}}_i) + \dot{{}^c \mathbf{t}}_e \quad (17)$$

Recall that ${}^c \dot{\mathbf{R}}_e {}^c \mathbf{R}_e^T = [{}^c \boldsymbol{\omega}_c]_{\times}$, where ${}^c \boldsymbol{\omega}_c$ is the rotational velocity expressed in the camera frame and $[\]_{\times}$ represents the skew-symmetric matrix associated to the vector cross-product. Hence, (17) can be written as follows:

$$\dot{q}_i {}^c \underline{\mathbf{u}}_i + q_i \dot{{}^c \underline{\mathbf{u}}_i} = [{}^c \boldsymbol{\omega}_c]_{\times} ({}^c \mathbf{t}_e - {}^c \mathbf{A}_i - q_i {}^c \underline{\mathbf{u}}_i) + \dot{{}^c \mathbf{t}}_e \quad (18)$$

Moreover, it is known that:

$${}^c \boldsymbol{\omega}_c \times {}^c \mathbf{t}_e + \dot{{}^c \mathbf{t}}_e = \dot{{}^c \mathbf{t}}_c \quad (19)$$

It follows that (20) can also be written as follows:

$$\dot{q}_i {}^c \underline{\mathbf{u}}_i + q_i \dot{{}^c \underline{\mathbf{u}}_i} = [\mathbf{I}_3 \quad [{}^c \mathbf{A}_i]_{\times} + q_i [{}^c \underline{\mathbf{u}}_i]_{\times}] {}^c \mathbf{v}_c \quad (20)$$

where ${}^c \mathbf{v}_c = [{}^c \dot{\mathbf{t}}_c \quad {}^c \boldsymbol{\omega}_c]^T$ is the Cartesian velocity of the camera frame, considered as attached to the base frame and moving with respect to a fixed end-effector, expressed in itself. Now, since ${}^c \underline{\mathbf{u}}_i$ is unit vector, it satisfies:

$$\begin{cases} {}^c \underline{\mathbf{u}}_i^T {}^c \underline{\mathbf{u}}_i = 1 \\ {}^c \underline{\mathbf{u}}_i^T \dot{{}^c \underline{\mathbf{u}}_i} = 0 \end{cases} \quad (21)$$

By projecting (20) on ${}^c \underline{\mathbf{u}}_i$ and combining the result with (21), one easily obtains the differential inverse kinematic model [2]:

$$\dot{\mathbf{q}} = {}^c \mathbf{J}_c^{inv} {}^c \mathbf{v}_c \quad (22)$$

$${}^c \mathbf{J}_c^{inv} = - \begin{bmatrix} {}^c \underline{\mathbf{u}}_1^T ({}^c \mathbf{A}_1 \times {}^c \underline{\mathbf{u}}_1)^T \\ \vdots \\ {}^c \underline{\mathbf{u}}_6^T ({}^c \mathbf{A}_6 \times {}^c \underline{\mathbf{u}}_6)^T \end{bmatrix} \quad (23)$$

Notice that ${}^c \mathbf{J}_c^{inv}$ can be considered as the interaction matrix associated to the joint values and thus be noted:

$${}^c \mathbf{J}_c^{inv} = \mathbf{L}_q \quad (24)$$

C. Visual servoing of leg directions

In few words, we recall that the time variation $\dot{\mathbf{s}}$ of the visual features \mathbf{s} can be expressed linearly with respect to the relative camera-object kinematics twist \mathbf{v} by $\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}$ where \mathbf{L}_s is the interaction matrix related to \mathbf{s} . The control scheme is usually designed to ensure an exponential decoupled decrease of the visual features to their desired value \mathbf{s}^* , from which we deduce if the object is motionless:

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}}_s^+ (\mathbf{s} - \mathbf{s}^*) \quad (25)$$

where $\widehat{\mathbf{L}}_s$ is a model or an approximation of \mathbf{L}_s , $\widehat{\mathbf{L}}_s^+$ the pseudo-inverse of $\widehat{\mathbf{L}}_s$, λ a positive gain tuning the time to convergence. To servo the leg directions, we define \mathbf{s} as the geodesic error between the current and desired legs orientation (${}^c \underline{\mathbf{u}} \times {}^c \underline{\mathbf{u}}^*$):

$$\mathbf{s}_{\mathbf{u}_i} = {}^c \underline{\mathbf{u}}_i \times {}^c \underline{\mathbf{u}}_i^*, i = 1..6 \quad (26)$$

This means that:

$$\mathbf{s}_{\mathbf{u}_i}^* = \mathbf{0}_{3 \times 1}, i = 1..6 \quad (27)$$

Using a similar demonstration as in the above subsection, by projecting (20) orthogonally to ${}^c \underline{\mathbf{u}}_i^T$ rather than on ${}^c \underline{\mathbf{u}}_i^T$,

the interaction matrix associated to a leg orientation ${}^c\mathbf{u}$ is obtained [2]:

$${}^c\dot{\mathbf{u}}_i = \mathbf{M}_i^T {}^c\mathbf{v}_c \quad (28)$$

$$\mathbf{M}_i^T = -\frac{1}{q_i} (\mathbf{I}_3 - {}^c\mathbf{u}_i {}^c\mathbf{u}_i^T) [\mathbf{I}_3 - [{}^c\mathbf{A}_i + q {}^c\mathbf{u}_i]_{\times}] \quad (29)$$

By combining (28) and (26), the time derivative of $\mathbf{s}_{\mathbf{u}_i}$ is given by:

$$\dot{\mathbf{s}}_{\mathbf{u}_i} = \mathbf{L}_{\mathbf{u}_i}^T {}^c\mathbf{v}_c \quad (30)$$

$$\mathbf{L}_{\mathbf{u}_i}^T = -[{}^c\mathbf{u}_i^*]_{\times} \mathbf{M}_i^T \quad (31)$$

Now, the standard method applies: we stack each individual errors in a single over-constrained vector $\mathbf{s}_{\mathbf{u}}$ and each associated individual interaction matrices $\mathbf{L}_{\mathbf{u}_i}^T$ into a compound one $\mathbf{L}_{\mathbf{u}}^T$ and impose a first-order convergence to $\mathbf{s}_{\mathbf{u}}$. This yields the following pseudo-control vector ${}^c\mathbf{v}_c$

$${}^c\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_{\mathbf{u}}^T \widehat{\mathbf{s}}_{\mathbf{u}} \quad (32)$$

By combining (22) and (32), the final control signal on the robot actuators is given by:

$$\dot{\mathbf{q}} = -\lambda {}^c\mathbf{J}_c^{inv} \widehat{\mathbf{L}}_{\mathbf{u}}^T \widehat{\mathbf{s}}_{\mathbf{u}} \quad (33)$$

where the “hat” notation means “estimated at each sample time from the measurements”.

IV. EXPERIMENTAL RESULTS

In this section, we first give a simple method to detect the edges of the platform legs from images. The direction of the platform legs are thus computed from the legs limbs in image. In the second paragraph the leg directions will be used in a visual-servo scheme of the platform effector position as described above.

A. Fast and automatic detection of the platform legs in images

Whatever visual servoing technique used, the spatiotemporal tracking of the visual information is the key of its success. This problem has been extensively studied in the case of perspective projection. In the literature, few works are concerned with the tracking problem in catadioptric sensor. For instance Barreto [4] dealt with straight lines tracking by contour-to-point tracker. [9] gave methods to obtain fast extraction and estimation algorithms of line for catadioptric image. Then, they explained how classical edge-tracking algorithms can be adapted to catadioptric sensors. Recently, a model based line tracking is given in [8]. Unfortunately, this kind of methods requires a good initialization of line projection in image. This initialization is often done manually. Furthermore, the line projection in each new image is determined by iterative method from its last position to its current position. Those methods suppose thus that the motion of the line projection in image is small enough to ensure the convergence. Parallel robots are supposed able to realize a large displacement in a limited period of time. Thus, the motion of the legs projection in image could

be very large. At this level the tracking algorithms based on iterative minimization might break down. To overcome those problems, we propose an automatic detection of the platform legs from an omnidirectional image. It is based on the particular position of the legs with respect to the camera. The projection of the legs in the image are almost radials (see Figure 4.a). This property is used to develop a fully automatic detection as follow:

- A set of circles with diameters ranging from a minimal value d_{min} to a maximal value d_{max} are considered. As we can see on Fig 4.a, d_{min} and d_{max} and the circle center are fixed such that only the image part, where the legs are projected is concerned. For the image given on Fig 4.a a set of 17 circles with $d_{min} = 184 \text{ pixel}$ and $d_{max} = 370 \text{ pixel}$ are defined.
- The image is scanned along each defined circle. We then get a mono-dimensional signal corresponding to each circle. For example, Figure 4.b gives the signal corresponding to the circle defined in Fig. 4.a.
- A binary signal (see 4.c) can be obtained after thresholding of the signal obtained in the above step (see Figure 4.b).
- The signal derivative after thresholding can be obtained using simple gradient filter. The derivative of the signal given on 4.c is plotted on Fig 4.d. The peaks of the signal define then the image of the leg limbs. It is possible to detect the peaks from the derivative of the signal given on 4.b without thresholding step (see Fig 4.e). However, from the last Figure we can note that other peaks appear. The thresholding step has got to avoid them and make the detection of the peaks belonging to the platform legs easier.

In conclusion of this paragraph, the method we propose allows an automatic detection of legs limbs in images. No initialization of the algorithm by the user is required. Furthermore, the legs detection in image does not depend on their velocity. In practice, our method take less than $0.3ms$ using a simple computer. It is then very adequate to high speed task. In the next paragraph, the legs projection will be used to compute their orientation in the camera frame.

B. Estimation of the legs orientation and their related interaction matrix

Assume now that the points belonging to legs limbs in image have been extracted using the method described in the above paragraph. The corresponding points in the normalized plane can then be obtained using the inverse of the transformation given by equation (4). Once the later are computed, the perpendicular vector to the interpretation plane can be computed by two ways:

- The conics parameters β_i defined in (10) can be computed using a linear minimization. From β_i , it is indeed possible to compute the perpendicular vector to the interpretation plane using (11).
- The camera has been calibrated. The point projections on the sphere defined by the unified camera model are

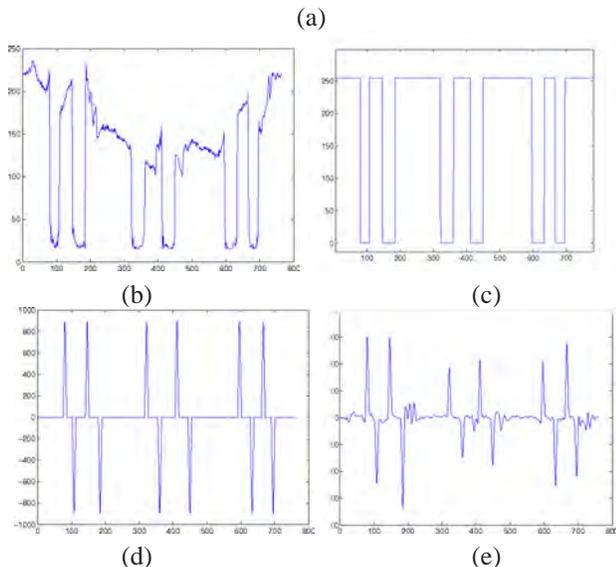
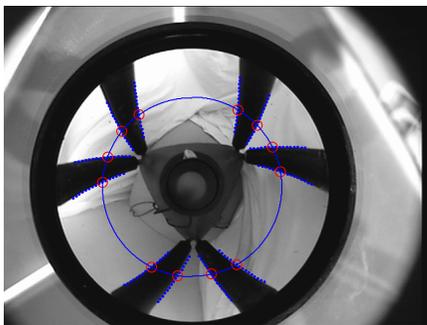


Fig. 4. Automatic detection of legs in image: (a) detection principle, (b) mono-dimensional signal along the defined circle, (c) the signal on circle after thresholding, (d) the derivative of the obtained signal after thresholding, (e) the derivative of the obtained signal without thresholding.

obtained from the point coordinates in the normalized plane and (5). The perpendicular vector to the interpretation plane of the legs limbs can be determined by linear solution using equation (7).

In practice, the second method gives more robust results with respect to noise. This was expected since the first method uses a non minimal parameters number (five parameters instead of only two independent ones). Indeed it estimates the five parameters of each conic using a linear method. While the second one uses a minimal parameters number in a linear optimization. Then, the later is used in the following of our experiment. Once the perpendicular vectors to the two leg limbs are computed, the corresponding leg orientations can be computed from (13).

Now, as the legs orientation are estimated, the numerical values of their related interaction matrices have also to be determined to close the control loop. From equation (28), we note that the later depend on the leg orientation, the attachment points \mathbf{A}_i expressed in the camera frame, the articulation value q_i and the legs orientation vector themselves. The joint values q_i appear two times in (28): under

the form ${}^c\mathbf{A}_i + q_i {}^c\mathbf{u}_i$ and as a gain. Considering the order of magnitude of \mathbf{A}_i and q_i , one can neglect small errors on the joint offsets. Moreover, since the joint are prismatic it is easy to measure their offsets manually with millimetric accuracy. This is also sufficient to ensure that the gain is accurate enough. Now, to totally determine the interaction matrices value, only the value of the attachment points \mathbf{A}_i have to be computed. In [2], a calibration procedure was proposed, using leg observation. The proposed method determine in first step the points \mathbf{A}_i expressed in the camera frame, then expressed in the basis frame. Finally, the other kinematic parameters can be deduced. This method can be combined with the automatic legs detection to make it more practical. The points \mathbf{A}_i can also be determined manually. In the next paragraph, a manually estimation of them will be enough to ensure convergence.

C. Example of an experimental result

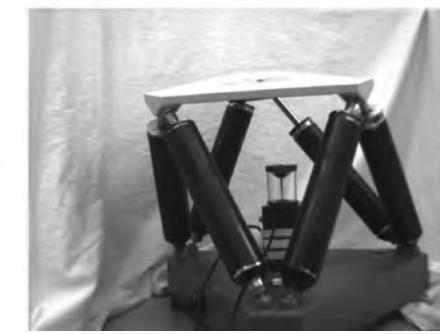
In the following experiment, we give an example of an omnidirectional visual-servo of the Gough-Stewart platform. The initial and desired configurations of the platform are given respectively on Figures 5.a and 5.b. The corresponding image are given respectively on 5.c and 5.d. In 5.d, both initial and desired positions of the legs limbs in image are given. In fact, in this experiment, the leg lengths are almost in their minimal possible values at the initial position. While, in desired position, the leg lengths are in their maximal values. Finally, 6.a gives the behaviors of the feature error squares $\mathbf{e}_i^T \mathbf{e}_i$. From this Figure we note that this error decrease to 0. A similar behavior of the velocities is also obtained on Figure 6.b. Furthermore, from Figure 6.c, we note the good behavior of the legs length to their desired values.

V. CONCLUSIONS AND FUTURE WORKS

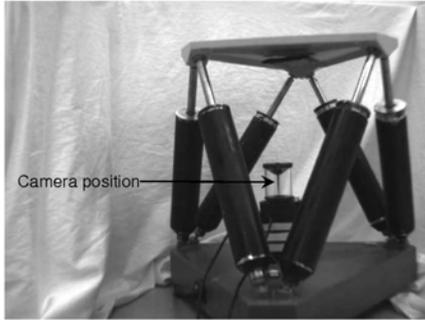
A novel approach was proposed for controlling a parallel robot using an omnidirectional camera. The use of omnidirectional camera allowed us to enlarge the field of view. Indeed, compared to classical camera, the omnidirectional camera allowed to observe the whole platform legs without any occlusion. The visibility constraint is ensured whatever the effector position. Furthermore, the legs positions with respect to the image plane made their detection by a fully automatic method very easy. No initialization of the legs positions in image is required. Furthermore, the proposed automatic detection method is very simple to implement and not time expensive. Then, it is very suitable for high speed task. Now, As the visibility constraint is ensured, future works will be devoted to prove the global convergence of the system. We are also interested to extend the results we obtain in this work to a new parallel robot namely T3R1

REFERENCES

- [1] N. Andreff, T. Dallej, and P. Martinet. Image-based visual servoing of a gough-stewart parallel manipulator using leg observations. *International Journal of Robotics Research*, 26(7):677–687, 2007.
- [2] N. Andreff and P. Martinet. Unifying kinematic modeling, identification and control of a gough-stewart parallel robot into a vision-based framework. *IEEE Transactions on Robotics*, 22(6):1077–1086, 2006.



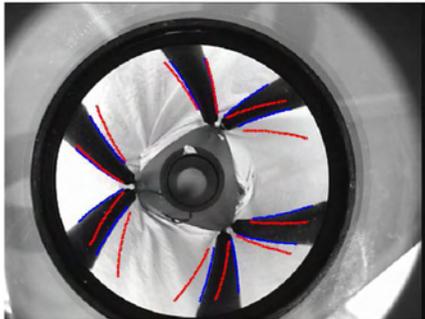
(a)



(b)



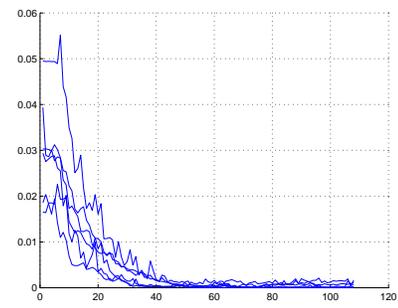
(c)



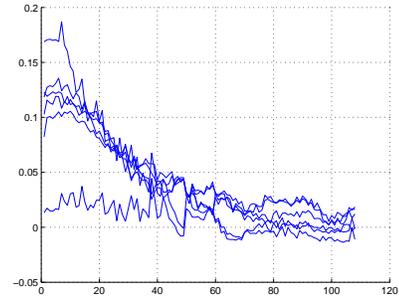
(d)

Fig. 5. Experimental results: (a) initial configuration, (b) desired configuration, (c) initial image, (d) desired image

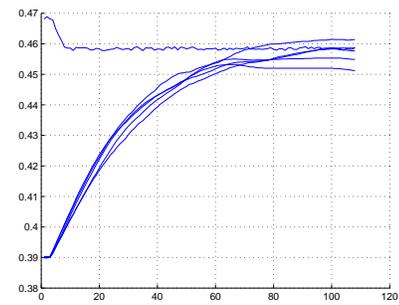
- [3] S. Baker and S. Nayar. A theory of catadioptric image formation. *International Journal of Computer Vision*, 35(2):175–196., November 1999.
- [4] F. M. J. Barreto and R. Horaud. Visual servoing/tracking using central catadioptric cameras. In P. D. B. Siciliano, editor, *International Symposium on Experimental Robotics*, Advanced Robotics Series. Springer-Verlag, July 2002.



(a)



(b)



(c)

Fig. 6. Experimental results: (a) errors behavior, (b) velocities, (c) Leg lengths behavior

- [5] J. P. Barreto and H. Araujo. Issues on the geometry of central catadioptric image formation. In *Computer Vision and Pattern Recognition*, volume 2, pages 422–427, Hawaii, USA, December 2001.
- [6] C. Geyer and K. Daniilidis. Mirrors in motion: Epipolar geometry and motion estimation. *International Journal on Computer Vision*, pages 766–773, 2003.
- [7] H. A. J. P. Barreto. Geometric properties of central catadioptric line images and their application in calibration. *IEEE Pattern Analysis and Machine Intelligence*, 27(8):1327–1333, August 2005.
- [8] E. Marchand and F. Chaumette. Fitting 3d models on central catadioptric images. In *IEEE International Conference on Robotics and Automation*, pages 52–58, Roma, April 2007.
- [9] C. Mei and E. Malis. Fast central catadioptric line extraction, estimation, tracking and structure from motion. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4774–4779, Beijing, October 2006.
- [10] C. Mei and P. Rives. Single view point omnidirectional camera calibration from planar grids. In *IEEE International Conference on Robotics and Automation*, Roma, April 2007.
- [11] T. Svoboda and T. Pajdla. Epipolar geometry for central catadioptric cameras. *International Journal on Computer Vision*, 49(1):23–37, August 2002.
- [12] Y. M. T. Dallej, N. Andreff and P. Martinet. 3d pose visual servoing

relieves parallel robot control from joint sensing. In *IROS 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4291–4296, Beijing, China, October 2006.