

IMAGE-BASED VISUAL SERVOING OF GOUGH-STEWART PARALLEL MANIPULATORS USING LEGS OBSERVATION

Nicolas Andreff* Tej Dallej* Philippe Martinet*

* *LASMEA-UMR 6602, CNRS-Université Blaise Pascal,
24 avenue des Landais, F-63177 Aubière Cedex,
firstname.lastname@lasmea.univ-bpclermont.fr,
<http://www.lasmea.univ-bpclermont.fr/Control>*

Abstract: In this paper, a tight coupling between computer vision and parallel robotics is exhibited through the projective line geometry. Indeed, contrary to the usual methodology where the robot is modeled independently from the control law which will be implemented, we take into account, since the early modeling stage, that vision will be used for control. Hence, kinematic modeling and projective geometry are fused into a control-devoted projective kinematic model. Thus, a novel vision-based kinematic modeling of a Gough-Stewart manipulator is proposed through the image projection of its cylindrical legs. Using this model, a visual servoing scheme is presented, where the image projection of the non-rigidly linked legs are servoed, rather than the end-effector pose. *Copyright ©2006 IFAC*

Keywords: Vision-based control, Parallel mechanisms, Line geometry

1. INTRODUCTION

There exists a large amount of work on the control of parallel mechanisms¹. In the focus of attention, Cartesian control is naturally achieved through the use of the inverse differential kinematic model which transforms Cartesian velocities into admissible joint velocities, taking into account the kinematic closures and avoiding to generate high internal forces (Dasgupta and Mruthyunjaya, 1998). It is noticeable that the inverse differential kinematic model of parallel mechanisms does not only depend on the joint configuration (as for serial mechanisms) but also on the end-effector pose.

Consequently, one needs to be able to estimate or measure the latter. As far as we know, all



Fig. 1. A Gough-Stewart platform.

the effort has been put on the estimation of the end-effector pose through the forward kinematic model and the joint measurements. However, this yields much trouble, related to the fact that there is usually no closed-form solution to the forward kinematic problem for parallel mechanisms. Hence, one numerically inverts the inverse kinematic model, which is analytically defined for most of the parallel mechanisms. However, it is known (Merlet, 1990; Husty, 1996) that this nu-

¹ See <http://www-sop.inria.fr/coprin/equipe/merlet> for a long list of references.

merical inversion requires high order polynomial root determination, with several possible solutions (up to 40 real solutions (Dietmaier, 1998) for a Gough-Stewart platform (Gough and Whitehall, 1962; Stewart, 1965)). Much of the work is thus devoted to solving this problem accurately and in real-time (see for instance (Zhao and Peng, 2000)), or to designing parallel mechanisms with analytical forward kinematic model (Kim and Tsai, 2002; Gogu, 2004). Alternately, one of the promising paths lies in the use of the so-called metrological redundancy (Baron and Angeles, 1998), which simplifies the kinematic models by introducing additional sensors into the mechanism and thus yields easier control (Marquet, 2002).

Vision being an efficient way of estimating the end-effector pose, it is a good alternative to use it for Cartesian control of parallel mechanisms. It can be done in two appropriate manners. The first one consists in measuring the end-effector pose by vision, and then performing 3D pose visual servoing (Martinet *et al.*, 1996) as in (Koreichi *et al.*, 1998; Kino *et al.*, 1999; Kallio *et al.*, 2000) (for parallel robots with a reduced number of degrees of freedom). However, this approach consists solely in a simple adaptation of now classical control schemes, which, although efficient, are not very innovative. Moreover, the direct application of visual servoing techniques assumes implicitly that the robot inverse differential kinematic model is given and that it is calibrated. Therefore, modeling, identification and control have small interaction with each other. Indeed, the model is usually defined for control using proprioceptive sensing only and does not foresee the use of vision for control, then identification and control are defined later on with the constraints imposed by the model. This is useful for modularity but this might not be efficient in terms of accuracy as well as of experimental set-up time.

On the opposite, a second approach was introduced in (Andreff *et al.*, 2005), where modeling, identification and control are performed knowing in advance that vision will be used at control time. To do so, the legs of the mechanism are servoed rather than its end-effector pose, which is useful in some practical cases. For instance, it is not wise to imagine observing the end-effector of a machining tool. On the opposite, it should not be a problem to observe the legs of the mechanism, even in such extreme cases.

In that method, the legs orientation were chosen as visual primitives and control was derived based on their reconstruction from the image. Although this reconstruction step consists only in computing the intersection of the two cylinder edges in the image, it might not be very accurate

for intrinsically geometrical reasons. Indeed, if a leg is parallel to the image, its edges will appear as parallel lines in the image and their intersection will lie at the infinite. Thus, in a close case to this one, the reconstruction will not be highly accurate and will impair the control.

In practice, this case is rapidly encountered. Indeed, since tracking lines in the image might be hard in the presence of camera distortion, one would chose a rather long focal lens (6 mm is here a long focal). Then, to observe all the legs in the image, one would place the camera at some distance from the manipulator (say, 1 m away from a desktop Gough-Stewart manipulator). In such an easy to set-up case, the cylinder edges appear nearly parallel in the image and control becomes unstable.

Consequently, following the original idea of visually servoing the legs and the common habit in visual servoing to derive control laws as close as possible to the image space, we propose in this paper to servo the leg edges rather than the leg orientation.

The outline of the paper is as follows. Section 2 recalls the model used for lines, then uses it to express the Gough-Stewart platform kinematics in the (static) camera frame and finally, recalls some useful geometric properties associated to the fact that most parallel mechanisms have cylindrical legs. Section 3 introduces, under the cylindrical legs assumption, the novel control law, expressed in the image and using the apparent edges of the legs as visual primitives for control. Section 4 presents the first ever experimental results of visual servoing using legs observation in its two variants: the already published leg orientation-based control and the novel edge-based control.

2. MODELING

2.1 Line modeling

A line \mathcal{L} in space, expressed in the camera frame, is defined by its binormalized Plücker coordinates (Andreff *et al.*, 2002):

$$\mathcal{L} \equiv ({}^c\mathbf{u}, {}^c\mathbf{n}, {}^cn) \quad (1)$$

where ${}^c\mathbf{u}$ is the unit vector giving the spatial orientation of the line, ${}^c\mathbf{n}$ is the unit vector defining the so-called interpretation plane of line \mathcal{L} and cn is a non-negative scalar. The latter are defined by ${}^cn{}^c\mathbf{n} = {}^c\mathbf{P} \times {}^c\mathbf{u}$ where ${}^c\mathbf{P}$ is any point on the line. Notice that, using this notation, the well-known Plücker coordinates (Plücker, 1865) are the couple $({}^c\mathbf{u}, {}^cn{}^c\mathbf{n})$.

The projection of such a line in the image plane, expressed in the camera frame, has for characteristic equation:

$${}^c \underline{\mathbf{n}}^T c \mathbf{p} = 0 \quad (2)$$

where ${}^c \mathbf{p}$ are the coordinates in the camera frame of any point in the image plane, lying on the line.

With the intrinsic parameters \mathbf{K} , one can easily obtain the conversion from the line equation in the camera frame ${}^c \underline{\mathbf{n}}$ to the same in pixel coordinates ${}^p \underline{\mathbf{n}}$ and vice-versa:

$${}^p \underline{\mathbf{n}} = \frac{\mathbf{K}^{-T} c \underline{\mathbf{n}}}{\|\mathbf{K}^{-T} c \underline{\mathbf{n}}\|} \quad (3)$$

$${}^c \underline{\mathbf{n}} = \frac{\mathbf{K}^T {}^p \underline{\mathbf{n}}}{\|\mathbf{K}^T {}^p \underline{\mathbf{n}}\|} \quad (4)$$

Notice that for numerical reasons, one should use normalized pixel coordinates. Namely, let us define the pixel frame by its origin located at the image center (*i.e.* the intersection of the image diagonals) and such that the pixel coordinates vary approximately between -1 and +1, by dividing them by, say, the image horizontal dimension in pixels.

2.2 Vision-based kinematics of an hexapod

Consider the hexapod in Figure 1. It has 6 cylindrical legs of varying length $q_i, i \in 1..6$, attached to the base by spherical joints located in points \mathbf{A}_i and to the moving platform (end-effector) by spherical joints located in points \mathbf{B}_i .

Rather than using the standard scalar inverse kinematic model of such an hexapod given by

$$\forall i \in 1..6, \quad q_i^2 = \overrightarrow{\mathbf{A}_i \mathbf{B}_i}^T \overrightarrow{\mathbf{A}_i \mathbf{B}_i} \quad (5)$$

expressing that q_i is the length of vector $\overrightarrow{\mathbf{A}_i \mathbf{B}_i}$, it is preferable for the subsequent derivation to use the vector form, introduced as the *vision-based kinematics of the hexapod* expressed in the camera frame in (Andreff *et al.*, 2005):

$$q_i {}^c \underline{\mathbf{u}}_i = {}^c \mathbf{R}_e {}^e \mathbf{B}_i + {}^c \mathbf{t}_e - {}^c \mathbf{A}_i \quad (6)$$

where ${}^c \underline{\mathbf{u}}_i$ is the spatial orientation of the i th leg. From the inverse kinematic model, one easily obtains the differential inverse kinematic model:

$$\dot{\mathbf{q}} = {}^c \mathbf{J}_c^{inv} {}^c \tau_c \quad (7)$$

$${}^c \mathbf{J}_c^{inv} = - \begin{bmatrix} {}^c \underline{\mathbf{u}}_1^T ({}^c \mathbf{A}_1 \times {}^c \underline{\mathbf{u}}_1)^T \\ \vdots \\ {}^c \underline{\mathbf{u}}_6^T ({}^c \mathbf{A}_6 \times {}^c \underline{\mathbf{u}}_6)^T \end{bmatrix} \quad (8)$$

where ${}^c \tau_c$ is the Cartesian velocity of the camera frame, considered as attached to the base frame and moving with respect to a fixed end-effector, and ${}^c \underline{\mathbf{u}}_i, i = 1..6$ are the unit vectors giving the pointing direction of each leg in the camera frame. Under the assumption that the legs are cylinders, those vectors can be easily detected as the intersection of the two cylinder edges in the image plane.

2.3 Cylindrical leg observation

Let us remark now that each cylinder edge is a line in space, with binormalized Plücker expressed in the camera frame (${}^c \underline{\mathbf{u}}_i, {}^c \underline{\mathbf{n}}_i^j, {}^c n_i^j$). Moreover, the attachment point \mathbf{A}_i is lying on the cylinder axis at distance R from the edge. Consequently, a cylinder edge is entirely defined by the following constraints, expressed here in the camera frame:

$${}^c \mathbf{A}_i^T {}^c \underline{\mathbf{n}}_i^j = -R \quad (9)$$

$${}^c \underline{\mathbf{n}}_i^j T {}^c \underline{\mathbf{n}}_i^j = 1 \quad (10)$$

$${}^c \underline{\mathbf{u}}_i^T {}^c \underline{\mathbf{n}}_i^j = 0 \quad (11)$$

3. EDGE-BASED VISUAL SERVOING

In the following subsection, the i subscript denoting the leg number was removed for clarity sake.

3.1 Interaction matrix

The interaction matrix \mathbf{N}^T relating the Cartesian velocity ${}^c \tau_c$ to the time derivative of the cylinder edges ${}^c \dot{\underline{\mathbf{n}}}^j$, expressed in the pixel frame:

$${}^c \dot{\underline{\mathbf{n}}}^j = \mathbf{N}^T {}^c \tau_c \quad (12)$$

can be decomposed into the product of three matrices:

$$\mathbf{N}^T = {}^p \mathbf{J}_c {}^h \mathbf{J}_u \mathbf{M}^T \quad (13)$$

From right to left, the first one is the interaction matrix associated to the leg orientation, it thus relates the time derivative of a leg orientation to ${}^c \tau_c$. The second transforms leg orientation velocities into leg edge velocities both expressed in the camera frame. Finally, the third one is associated to the camera-to-pixel change of frame. Below, the expression of the leg orientation interaction matrix is first recalled then the last two matrices are derived.

3.1.1. Leg orientation interaction matrix The control proposed in (Andreff *et al.*, 2005) servoed the geodesic error between the current and desired legs orientation (${}^c \underline{\mathbf{u}} \times {}^c \underline{\mathbf{u}}^*$) and thus introduced the interaction matrix associated to a leg orientation:

$${}^c \dot{\underline{\mathbf{u}}} = \mathbf{M}^T {}^c \tau_c \quad (14)$$

$$\mathbf{M}^T = -\frac{1}{q} (\mathbf{I}_3 - {}^c \underline{\mathbf{u}} {}^c \underline{\mathbf{u}}^T) [\mathbf{I}_3 - [{}^c \mathbf{A} + q {}^c \underline{\mathbf{u}}]_{\times}] \quad (15)$$

3.1.2. Edge velocity in the camera frame Let us first derive the time derivative of a cylinder edge, expressed in the camera frame, and under the kinematic constraint that the cylinder is attached to the base by a spherical or universal joint located in \mathbf{A} . To do so, let us differentiate (9)-(11):

$${}^c \dot{\mathbf{n}}^{jT} \mathbf{A} = 0 \quad (16)$$

$${}^c \dot{\mathbf{n}}^{jT} c \underline{\mathbf{n}}^j = 0 \quad (17)$$

$${}^c \dot{\mathbf{n}}^{jT} c \underline{\mathbf{u}} + {}^c \dot{\mathbf{n}}^{jT} c \underline{\mathbf{u}} = 0 \quad (18)$$

From (17) and the fact that $(\underline{\mathbf{u}}, \underline{\mathbf{n}}, \underline{\mathbf{u}} \times \underline{\mathbf{n}})$ form an orthonormal basis (Andreff *et al.*, 2002), one can state:

$${}^c \dot{\mathbf{n}}^j = \alpha c \underline{\mathbf{u}} + \beta c \underline{\mathbf{u}} \times c \underline{\mathbf{n}}^j \quad (19)$$

Inserting this expression into (16) and (18) yields

$$\alpha = -{}^c \dot{\mathbf{n}}^{jT} c \underline{\mathbf{u}}, \quad \beta = \frac{{}^c \mathbf{A}^T c \underline{\mathbf{u}}}{c \mathbf{A}^T (c \underline{\mathbf{u}} \times c \underline{\mathbf{n}}^j)} {}^c \dot{\mathbf{n}}^{jT} c \underline{\mathbf{u}} \quad (20)$$

Consequently, one obtains the relationship between the time derivative of a leg edge, expressed in the camera frame, and the time derivative of the leg orientation

$${}^c \dot{\mathbf{n}}^j = {}^h \mathbf{J}_u c \underline{\mathbf{u}} \quad (= {}^h \mathbf{J}_u \mathbf{M}^T c \tau_c) \quad (21)$$

$${}^h \mathbf{J}_u = - \left(\mathbf{I}_3 - \frac{(c \underline{\mathbf{u}} \times c \underline{\mathbf{n}}^j) c \mathbf{A}^T}{c \mathbf{A}^T (c \underline{\mathbf{u}} \times c \underline{\mathbf{n}}^j)} \right) c \underline{\mathbf{u}} c \underline{\mathbf{n}}^{jT} \quad (22)$$

3.1.3. Image line velocity in pixel coordinates

Let us now derive the Jacobian associated to the change of frame, where the time derivative of an image line is expressed, from the camera frame to the pixel frame. Note that this paragraph holds for any image line, not only for cylinder edges.

Rewriting (3) as

$${}^p \underline{\mathbf{n}} = \mu ({}^c \underline{\mathbf{n}}) \mathbf{K}^{-T} c \underline{\mathbf{n}} \quad (23)$$

we can differentiate the latter with time:

$${}^p \dot{\underline{\mathbf{n}}} = \frac{d\mu({}^c \underline{\mathbf{n}})}{dt} \mathbf{K}^{-T} c \underline{\mathbf{n}} + \mu ({}^c \underline{\mathbf{n}}) \mathbf{K}^{-T} c \dot{\underline{\mathbf{n}}} \quad (24)$$

Taking into account again that ${}^p \underline{\mathbf{n}}$ is a unit vector (17), one gets

$$\left(\frac{d\mu({}^c \underline{\mathbf{n}})}{dt} \mathbf{K}^{-T} c \underline{\mathbf{n}} \right)^T {}^p \underline{\mathbf{n}} + \mu ({}^c \underline{\mathbf{n}}) {}^p \underline{\mathbf{n}}^T \mathbf{K}^{-T} c \dot{\underline{\mathbf{n}}} = 0 \quad (25)$$

Using (3) again, this simplifies into

$$\frac{d\mu({}^c \underline{\mathbf{n}})}{dt} = -\mu ({}^c \underline{\mathbf{n}}) {}^2 p \underline{\mathbf{n}}^T \mathbf{K}^{-T} c \dot{\underline{\mathbf{n}}} \quad (26)$$

Inserting this result in (24) yields

$${}^p \dot{\underline{\mathbf{n}}} = (-\mathbf{K}^{-T} c \underline{\mathbf{n}}^T \mu ({}^c \underline{\mathbf{n}}) {}^2 p \underline{\mathbf{n}}^T + \mu ({}^c \underline{\mathbf{n}}) \mathbf{I}_3) \mathbf{K}^{-T} c \dot{\underline{\mathbf{n}}} \quad (27)$$

which simplifies into

$${}^p \dot{\underline{\mathbf{n}}} = \mu ({}^c \underline{\mathbf{n}}) (\mathbf{I}_3 - {}^p \underline{\mathbf{n}} {}^p \underline{\mathbf{n}}^T) \mathbf{K}^{-T} c \dot{\underline{\mathbf{n}}} \quad (28)$$

Introducing (4) in (23) proves that

$$\mu ({}^c \underline{\mathbf{n}}) = \|\mathbf{K}^T {}^p \underline{\mathbf{n}}\| \quad (29)$$

from which we finally obtain the relationship between the time derivative of a line, expressed in the image frame, and the same expressed in the camera frame

$${}^p \dot{\underline{\mathbf{n}}} = {}^p \mathbf{J}_c c \dot{\underline{\mathbf{n}}} \quad (= {}^p \mathbf{J}_c {}^h \mathbf{J}_u \mathbf{M}^T c \tau_c) \quad (30)$$

$${}^p \mathbf{J}_c = \|\mathbf{K}^T {}^p \underline{\mathbf{n}}\| (\mathbf{I}_3 - {}^p \underline{\mathbf{n}} {}^p \underline{\mathbf{n}}^T) \mathbf{K}^{-T} \quad (31)$$

3.2 Control

Since we want to drive the unit vectors associated to the leg edges to their desired values, we choose to servo the geodesic errors

$$\mathbf{e}_{i,j} = {}^p \underline{\mathbf{n}}_i^j \times {}^p \underline{\mathbf{n}}_i^{j*}, j = 1..2, i = 1..6 \quad (32)$$

whose time derivatives are

$$\dot{\mathbf{e}}_{i,j} = \mathbf{L}_{i,j}^T {}^p \dot{\underline{\mathbf{n}}}_i^j \quad (33)$$

$$\mathbf{L}_{i,j}^T = -[{}^p \underline{\mathbf{n}}_i^{j*}] \times \mathbf{N}_{i,j}^T \quad (34)$$

where $i = 1..6$ denotes the legs and $j = 1..2$ the edges.

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

$${}^c \tau_c = -\lambda \hat{\mathbf{L}}^T \mathbf{e} \quad (35)$$

which is fed to the actuators through the vision-based differential inverse kinematic model (8) to deliver the final control signal

$$\hat{\mathbf{q}} = -\lambda {}^c \mathbf{J}^{inv} \hat{\mathbf{L}}^T \mathbf{e} \quad (36)$$

where the $\hat{\cdot}$ means estimated.

Notice that this control makes use of the detected edges in the image, the joint values, the intrinsic parameters and the attachment points of the legs onto the base expressed in the camera frame. However, notice that neither the attachment points of the legs onto the mobile platform nor the radius of the legs are used here explicitly, which reduces the number of kinematic parameters to be calibrated.

4. RESULTS

The experimental robot has an analog joint position-controller that we interfaced with Linux-RTAI. Joint velocity control is emulated through this position-controller with an approximate² 20ms sampling period. Frame grabbing, line tracking and numerical computation are performed using ViSP, an open C++ library for visual servoing (Marchand *et al.*, 2005).

It also has to be noticed that the mechanism presents high friction disturbances that have not yet been compensated for since friction seems to

² This part is not yet running under RTAI, but only under standard Linux.

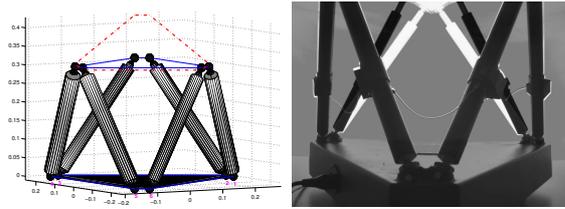


Fig. 2. Composition of the desired (black) and initial (white) configurations.

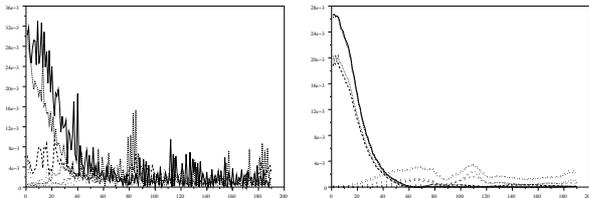


Fig. 3. Behavior of orientation-based control: Evolution of the controlled leg orientation errors (left) and of the non-controlled edge errors (right) with respect to time.

depend non trivially on the robot configuration. Hence, to overcome these disturbances, we implemented the visual servoing control with an adaptive gain, function of the controlled error norm: low at init, high near convergence.

The robot is observed by a camera with 4.8mm focal length, placed in front of it. Figure 2 (left) presents a schematic view of the initial (solid line) and desired (dotted line) configurations of the robot, while Figure 2 (right) is obtained by merging the (video inverted) view from the camera of the robot in the initial configuration and the view in the desired configuration. The initial configuration yields equal leg lengths (375mm) whereas in the desired configuration the two legs in the back are extended by an additional 100mm.

To show the robustness of the approach, we deliberately placed ourselves in a difficult configuration: approximate calibration of the robot and camera, camera placed 70cm away from the robot base center, two legs close to be in a frontal parallel configuration.

We display for both controls the evolution of both the controlled and non-controlled errors: in Figure 3 for orientation-based control and in Figure 4 for edge-based control. The desired exponential behavior is obtained, yet disturbed by friction and distorted by the adaptive gain strategy.

It is noticeable (Figure 3) that the orientation error signal is extremely noisy, as expected. Thus, orientation-based control tries its best to servo it to zero, but does not succeed in bringing the robot to the desired configuration since the edge errors do not reach zero. On the opposite (Figure 4), edge-based control succeeds (the edges are

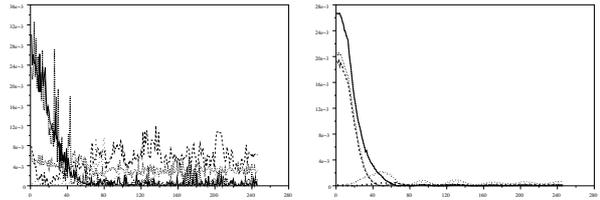


Fig. 4. Behavior of edge-based control: Evolution of the non-controlled leg orientation errors (left) and of the controlled leg edge errors (right) with respect to time.

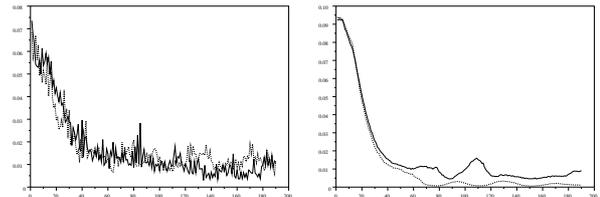


Fig. 5. Comparison of the behavior of orientation-based (solid line) and edge-based control (dotted line): evolution of the norm of the legs orientation error (left) and of the edges error (right) with respect to time.

almost aligned on their reference) even though the orientation signal is noisy and biased. Figure 5 presents a complementary view of this behavior by displaying the evolution of both the controlled and non controlled error norms: the orientation signal is very noisy and converges equally poorly in both controls (left) while the edge signal is much cleaner and converges only in edge-based control.

The reason for such a behavior is, as stated in the early introduction, that noise is appearing in the servoed error in orientation-based control while it only appears in the interaction matrix in edge-based control.

5. CONCLUSION

We extended previous results concerning (PBVS-like) leg orientation-based control of a Gough-Stewart to an (IBVS-like) edge-based visual servoing scheme. It benefits from the advantages of the orientation-based visual servoing of the Gough-Stewart legs: reduced calibration parameters set, low dependence on the joint values and ability to servo the robot even though the end-effector is not visible. However, we improved the practical robustness (although it still has to be proven theoretically) by servoing the legs in the image: almost all the calibration parameters (intrinsic parameters of the camera and base points) and numerical errors remain located in the interaction matrix.

To do so, we took advantage of a common use of line geometry in kinematics, vision and visual servoing. This allows for an optimal modeling of Gough-Stewart parallel robots, provided that vision is used at control time. This modeling was established under the hypothesis that the camera is calibrated, but this result might be extendable to the use of an uncalibrated camera. Nevertheless, this extension is not necessary since the control is done in the very projective space associated to image lines, while the reconstructed or calibrated Euclidean terms only appear in the interaction matrix where extreme accuracy is not required.

However, self-occlusions of the mechanism with respect to a single camera are still a matter of study, although the observation of edges should simplify the problem since the two edges of a given leg are seldom hidden simultaneously. A way to overcome the occlusion problem is to turn oneself to multi-camera perception systems.

ACKNOWLEDGMENT

This study was jointly funded by CPER Auvergne 2003-2005 program and by the CNRS-ROBEA program through the MP2 project. The authors warmly thank Eric Marchand for providing them with an early version of ViSP.

REFERENCES

- Andreff, N., A. Marchadier and P. Martinet (2005). Vision-based control of a Gough-Stewart parallel mechanism using legs observation. In: *Proc. Int. Conf. Robotics and Automation (ICRA'05)*. Barcelona, Spain. pp. 2546–2551.
- Andreff, N., B. Espiau and R. Horaud (2002). Visual servoing from lines. *Int. Journal of Robotics Research* **21**(8), 679–700.
- Baron, L. and J. Angeles (1998). The on-line direct kinematics of parallel manipulators under joint-sensor redundancy. In: *Advances in Robot Kinematics*. Kluwer Academic Publishers. pp. 126–137.
- Dasgupta, B. and T.S. Mruthyunjaya (1998). Force redundancy in parallel manipulators: theoretical and practical issues. *Mech. Mach. Theory* **33**(6), 727–742.
- Dietmaier, P. (1998). The Stewart-Gough platform of general geometry can have 40 real postures. In: *Advances in Robot Kinematics: Analysis and Control* (J. Lenarčič and M. L. Husty, Eds.). pp. 1–10. Kluwer.
- Gogu, G. (2004). Fully-isotropic T3R1-type parallel manipulator. In: *On Advances in Robot Kinematics* (J. Lenarčič and C. Galletti, Eds.). pp. 265–272. Kluwer Academic Publishers.
- Gough, V.E. and S.G. Whitehall (1962). Universal tyre test machine. In: *Proc. FISITA 9th Int. Technical Congress*. pp. 117–137.
- Husty, M. (1996). An algorithm for solving the direct kinematics of general Gough-Stewart platforms. *Mech. Mach. Theory* **31**(4), 365–380.
- Kallio, P., Q. Zhou and H. N. Koivo (2000). Three-dimensional position control of a parallel micromanipulator using visual servoing. In: *Microrobotics and Microassembly II, Proceedings of SPIE* (Bradley J. Nelson and Editors Jean-Marc Breguet, Eds.). Vol. 4194. Boston, USA. pp. 103–111.
- Kim, H.S. and L.-W. Tsai (2002). Evaluation of a Cartesian parallel manipulator. In: *Advances in Robot Kinematics: Theory and Applications* (J. Lenarčič and F. Thomas, Eds.). Kluwer Academic Publishers.
- Kino, H., C.C. Cheah, S. Yabe, S. Kawamura and S. Arimoto (1999). A motion control scheme in task oriented coordinates and its robustness for parallel wire driven systems. In: *Int. Conf. Advanced Robotics (ICAR'99)*. Tokyo, Japan. pp. 545–550.
- Koreichi, M.L., S. Babaci, F. Chaumette, G. Fried and J. Pontnau (1998). Visual servo control of a parallel manipulator for assembly tasks. In: *6th Int. Symposium on Intelligent Robotic Systems, SIRS'98*. Edimburg, Scotland. pp. 109–116.
- Marchand, E., F. Spindler and F. Chaumette (2005). ViSP: a generic software platform for visual servoing. *IEEE Robotics and Automation Magazine*.
- Marquet, F. (2002). Contribution à l'étude de l'apport de la redondance en robotique parallèle. PhD thesis. LIRMM - Univ. Montpellier II.
- Martinet, P., J. Gallice and D. Khadraoui (1996). Vision based control law using 3d visual features. In: *Proc. World Automation Congress, WAC'96, Robotics and Manufacturing Systems*. Vol. 3. Montpellier, France. pp. 497–502.
- Merlet, J-P. (1990). An algorithm for the forward kinematics of general 6 d.o.f. parallel manipulators. Technical Report 1331. INRIA.
- Plücker, J. (1865). On a new geometry of space. *Philosophical Transactions of the Royal Society of London* **155**, 725–791.
- Stewart, D. (1965). A platform with six degrees of freedom. In: *Proc. IMechE (London)*. Vol. 180. pp. 371–386.
- Zhao, X. and S. Peng (2000). Direct displacement analysis of parallel manipulators. *Journal of Robotics Systems* **17**(6), 341–345.