Unifying Kinematic Modeling, Identification, and Control of a Gough–Stewart Parallel Robot Into a Vision-Based Framework

Nicolas Andreff and Philippe Martinet

***Abstract*—In this paper, it is shown that computer vision, used as an exteroceptive redundant metrology mean, simplifies the control of a Gough–Stewart parallel robot. Indeed, contrary to the usual methodology, where the robot is modeled independently from the control law which will be implemented, we take into account that vision will be used for control, from the early modeling stage. Hence, kinematic modeling and projective geometry are fused into a control-devoted projective kinematic model. Thus, a novel vision-based kinematic modeling of such a robot is proposed through the observation of its legs. Inspired by the geometry of lines, this model unifies and simplifies both identification and control. Indeed, it has a reduced parameter set, and allows us to propose a linear solution to its calibration. Using the same model, a visual servoing scheme is presented, where the attitudes of the nonrigidly linked legs are servoed, rather than the end-effector pose. Finally, theoretical results concerning the stability of this control law are provided.**

***Index Terms*—Computer vision, line geometry, parallel kinematics, visual servoing.**

I. Introduction

PARALLEL mechanisms are such that there exist several kinematic chains (or legs) between their base and their moving platform (also called the end-effector below). Therefore, they may exhibit a better repeatability [1] than serial mechanisms, but not necessarily a better accuracy [2], because of the large number of links and passive joints. There can be two ways to improve the accuracy. The first way is to perform a kinematic calibration of the mechanism, and the second one is to use a control law which is robust to calibration errors.

There exists a large amount of work on the control of parallel mechanisms (see [3] for a long list of references). In the focus of attention, Cartesian control is naturally achieved through the use of the inverse differential kinematic model (abusively called the robot Jacobian, since it can not be expressed as a matrix of partial derivatives of a map with respect to its coordinates), which transforms Cartesian velocities into joint velocities. 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 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 analytic formulation of the forward kinematic model of a parallel mechanism. Hence, one numerically inverts the inverse kinematic model, which is algebraically defined for most of the parallel mechanisms. However, it is known [4], [5] that this numerical inversion requires high-order polynomial root determination, with several possible solutions (up to 40 real solutions [6] for a Gough–Stewart platform [7], [8]). Much of the work is thus devoted to solving this problem accurately and in real time (see, for instance, [9]), or to designing parallel mechanisms with algebraic forward kinematic model [10], [11]. Alternately, one of the promising paths lies in the use of the so-called metrological redundancy [12], which simplifies the kinematic models by introducing additional sensors into the mechanism, and thus yields easier control [13].

Computer vision being an efficient way of estimating the end-effector pose [14], [15], it is a good alternative to use it for Cartesian control of parallel mechanisms. It can be done in three ways.

1) Vision as a Sensor: The first one consists of computing the end-effector poses by vision, then in translating them into joint configurations through the inverse kinematic model, and finally servoing in the joint space. This scheme is rather easy to implement for serial mechanisms, provided that inverting the forward kinematic model can be done satisfactorily. The latter is straightforward for parallel mechanisms, since they usually have an algebraic inverse kinematic model. Similarly, one can consider computer vision as a contactless redundant sensor, as already stated in the context of parallel mechanism calibration [16], and use the simplified models based on the redundant metrology paradigm.

However, such schemes should be used carefully for parallel mechanisms, since joint control does not take into account the kinematic closures, and may, therefore, yield high internal forces [17]. Moreover, there may exist several end-effector poses associated with a given joint configuration. Hence, a simple joint control may converge to the wrong end-effector pose, even though it converges to the correct joint configuration.

Manuscript received January 18, 2006; revised April 20, 2006. This paper was recommended for publication by Associate Editor H. Zhuang and Editor F. Park upon evaluation of the reviewers' comments. This work was supported in part by the CPER Auvergne 2003–2005 program and in part by the European Community under the IP NEXT Project 011815. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005. Color versions of Figs. 3 and 7–11 are available at http://ieeexplore.org.

The authors are with the LASMEA–CNRS–Université Blaise Pascal/IFMA, 63175 Aubière, France (e-mail: andreff@lasmea.univ-bpclermont.fr; martinet@lasmea.univ-bpclermont.fr).

Digital Object Identifier 10.1109/TRO.2006.882931

2) Visual Servoing: Second, vision can be additionally used to perform visual servoing [18]. Indeed, instead of measuring the end-effector pose and converting it into joint values, one could think of using this measure directly for control. Recall that there exist many visual servoing techniques, ranging from position-based visual servoing (PBVS) [19] (when the pose measurement is explicit) to image-based visual servoing (IBVS) [18] (when it is made implicit by using only image measurements). Most applications embed the vision system onto the end-effector to position the latter with respect to a rigid object whose accurate position is unknown, but one can also find applications with a fixed camera observing the end-effector [20]. The interested reader is referred to [21] for a thorough and up-to-date state-of-the-art.

Visual servoing techniques are very effective, since they close the control loop over the vision sensor. This yields a high robustness to perturbations as well as to calibration errors. Indeed, these errors only appear in a Jacobian matrix, but not in the regulated error.

Essentially, visual servoing techniques generate a Cartesian desired velocity, which is converted into joint velocities by the robot inverse differential kinematic model. Hence, one can translate such techniques to parallel mechanisms, as in [22]–[24] [for parallel robots with a reduced number of degrees of freedom (DOFs)], by observation of the robot end-effector and the use of standard kinematic models. It is rather easier than in the serial case, since the inverse differential kinematic model of a parallel mechanism is usually algebraic. Moreover, for parallel mechanisms, since the joint velocities are filtered through the inverse differential kinematic model, they are admissible, in the sense that they do not generate internal forces. More precisely, this is only true in the ideal case. However, if the estimated inverse differential kinematic model used for control is close enough to the actual one, the joint velocities will be closely admissible, in the sense that they do not generate high internal forces. The only difficulty for end-effector visual servoing of a parallel mechanism would come from the dependency of the inverse differential kinematic model to the Cartesian pose, which would need be estimated, but, as stated above, vision can also do that [14], [15]! Notice that this point pleads for PBVS rather than IBVS of parallel mechanisms, which is effectively the choice made in [22]–[24].

From the above discussion, we thus highly recommend using visual servoing for parallel mechanism control.

3) A Novel Approach: However, the previous two ways consist solely of a simple adaptation of now classical control schemes, which, although probably very 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 (Fig. 1). This is useful for modularity, but this might not be efficient in terms of accuracy, as well as in experimental setup time.

On the opposite side, a unified framework for modeling, identification, and control, apart from being definitely more satisfying for the mind, would certainly open a path to higher efficiency. Indeed, instead of having identification and control being driven by the initial modeling stage, one could have a model taking into account the use of vision for control, and hence, for identification. To do so, it is necessary to fuse robot kinematics and projective geometry into a projective kinematic model (Fig. 2). Thus, we propose a novel third way to use vision, which gathers the advantages of redundant metrology and of visual servoing and avoids most of their drawbacks.

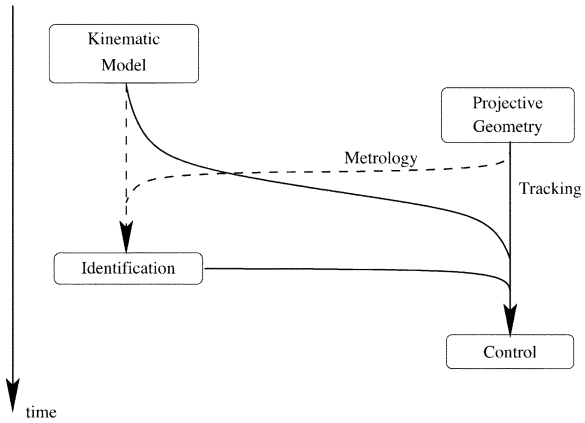


Fig. 1. Usual cascade from modeling to vision-based control.

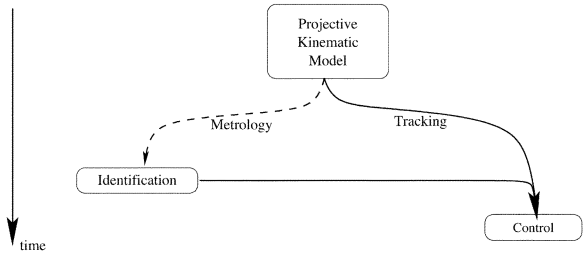


Fig. 2. Simplified cascade from modeling to vision-based control using a projective kinematic model.

Moreover, observing the end-effector of a parallel mechanism by vision may be incompatible with its application. For instance, it is not wise to imagine observing the end-effector of a machining tool. On the opposite side, it should not be a problem to observe the legs of the mechanism, even in such extreme cases. Thereby one would turn vision from an exteroceptive sensor to a somewhat more proprioceptive sensor. This brings us back to the redundant metrology paradigm.

Parallel mechanisms are most often designed with slim and rectilinear legs. Thus, one is inclined to consider them as straight lines, as it was done for their kinematic analysis [1], [5] or kinematic calibration [16], [25]–[27]. Therefore, the line geometry [28], [29] is certainly the heart of the unification, all the more as line geometry is widely used in kinematic analysis [30], [31] and computer vision [32], and has already been used for visual servoing [33]–[35].

Previous work on kinematic calibration [25]–[27] already considered vision as a way to deliver contactless metrological redundancy. However, with the exception of [27], the models that were calibrated remain classical. Indeed, vision was only used for sensing, and neither modeling nor control was

Fig. 3. Experimental set-up: A Gough–Stewart platform observed by a camera.

questioned from the vision point of view. A first step in this direction was made in [36], where vision was used already at the modeling stage in order to derive a visual servoing scheme based on the observation of a Gough–Stewart parallel robot [7], [8]. However, full analysis of the model and its calibration was not addressed.

Consequently, the contribution of this paper is to present an original and unifying vision-based modeling *and* identification *and* control framework of parallel mechanisms by observing their legs with a camera fixed with respect to the base. This framework, inspired by line geometry, synthesizes into a single convergent framework previous work on identification [37] and on control [36], simplifying identification and going deeper into control analysis. It is introduced in the case of parallel mechanisms of the hexapod type, with illustration on a Gough–Stewart platform (Fig. 3).

The remainder of the paper is the following. Section II emphasizes the vision-based kinematics of the hexapod, which is the key point of the paper. A discussion of this model with respect to sensing follows in Section III. Then, Section IV recalls the differential geometry aspect of the leg observation and the control derived from it. Section V contains a closed-form linear solution to the identification of the model used for control. Finally, simulation and experimental results and the conclusion are given, respectively, in Sections VI and VII.

II. Vision-Based Kinematics

A. Preliminaries

This paper assumes that the vision system is calibrated, which is not a strong hypothesis anymore, since accurate and easy-to-use camera calibration software can easily be found on the Web [38]. The classical approach for calibrating a camera is to use a calibration grid made of points [15], [39]–[41]. However, in this case, where we aim at a unifying framework using line geometry, it may be more satisfactory for the mind to use the method proposed in [42]. Indeed, this method is based on the observation of a set of lines, without any assumption on their 3-D structure. As can be seen in Fig. 3, there are plenty of lines in the images observed by a camera placed in front of a Gough–Stewart parallel robot. Hence, the camera may be calibrated *online* without any experimental burden.

Since we plan to use line geometry as a central element for modeling, a representation for lines suited to control and identification is needed. Among the work on visual servoing from lines [18], [33]–[35], [43], [44], we prefer the so-called *binormalized Plücker coordinates* representation in [34], which turns out to be coherent with kinematic modeling of parallel mechanisms.

In such a representation, a straight line in the oriented 3-D space [45] is modeled by the triplet $(\underline{\mathbf{u}}, \underline{\mathbf{h}}, \mathrm{h})$ where:

- $\underline{\mathbf{u}}$ is the unit vector giving the spatial direction of the line;
- $\underline{\mathbf{h}}$ is also a unit vector, and h is a nonnegative scalar. They are defined by the cross-product $h\underline{\mathbf{h}} = \overrightarrow{\mathbf{OP}} \times \underline{\mathbf{u}}$, where $\mathbf{P}$ is any point on the line and depends on the reference point $\mathbf{O}$.

Notice that using this notation, the well-known (normalized) Plücker coordinates [28], [31] are the couple $(\underline{\mathbf{u}}, h\underline{\mathbf{h}})$.

An interesting property of this representation, concerning computer vision, is that if $\mathbf{O}$ is chosen as the center of projection, then $\underline{\mathbf{h}} = (h_x, h_y, h_z)^T$ represents the image projection of the line, i.e., the equation of the image line verifies

$$h_x x + h_y y + h_z = 0 \tag{1}$$

where x and y are the coordinates of a point in the image. The interpretation of the scalar h is the orthogonal distance of the line to the center of projection.

Another interesting property of this representation is that for any point $\mathbf{P} = (X, Y, Z)^T$ in the 3-D space, the dot product $\underline{\mathbf{h}}^T\mathbf{P}$ is the signed orthogonal distance of $\mathbf{P}$ to the interpretation plane (i.e., the plane containing the projection center and the image line), defined by $\underline{\mathbf{h}}$.

B. Kinematics of an Hexapod

Consider the hexapod in Fig. 3. It has six legs of varying length $q_i, i \in 1, \ldots, 6$, 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 $\mathbf{B}_i$.[1] The inverse kinematic model of such an hexapod is given by

$$\forall i \in 1, \ldots, 6, \quad q_i^2 = \overrightarrow{\mathbf{A}_i\mathbf{B}_i}^T \overrightarrow{\mathbf{A}_i\mathbf{B}_i} \tag{2}$$

expressing that q_i is the length of vector $\overrightarrow{\mathbf{A}_i\mathbf{B}_i}$. This model can be expressed in any Euclidean reference frame. Hence, it can be expressed in the base frame $\mathcal{R}_b$, in the end-effector frame $\mathcal{R}_e$, or in the camera frame $\mathcal{R}_c$. In the remainder and when needed, the reference frame used will be made explicit by a left superscript.

Let us consider $(\underline{\mathbf{u}}_i, \underline{\mathbf{h}}_i, h_i)$, the binormalized Plücker coordinates of the line passing through $\mathbf{A}_i$ and $\mathbf{B}_i$, oriented from $\mathbf{A}_i$ to $\mathbf{B}_i$. Then, one trivially has

$$\overrightarrow{\mathbf{A}_i\mathbf{B}_i} = q_i\underline{\mathbf{u}}_i \tag{3}$$

$$\mathbf{A}_i \times \underline{\mathbf{u}}_i = \mathbf{B}_i \times \underline{\mathbf{u}}_i = h_i\underline{\mathbf{h}}_i. \tag{4}$$

[1]Notice that the spherical joints imply an uncontrollable rotation of the legs around their axis, which has no effect on the end-effector position.

From [1], it is known that the inverse differential kinematic model of the hexapod, relating the end-effector Cartesian velocity (i.e., its kinematic twist)

$$\tau_e = \begin{pmatrix} V_e \\ \Omega_e \end{pmatrix}$$

to the joint velocities is

$$\mathbf{D}_e^{\text{inv}} = \begin{pmatrix} \underline{\mathbf{u}}_1^T & (\overrightarrow{\mathbf{CB}}_1 \times \underline{\mathbf{u}}_1)^T \\ \vdots & \vdots \\ \underline{\mathbf{u}}_6^T & (\overrightarrow{\mathbf{CB}}_6 \times \underline{\mathbf{u}}_6)^T \end{pmatrix} \tag{5}$$

where $\mathbf{C}$ is the center of the end-effector reference frame. Notice that the inverse differential kinematic model is written $\mathbf{D}_e^{\text{inv}}$ rather than $\mathbf{J}^{-1}$, to clearly state that it has an algebraic expression, contrary to the inverse differential kinematic model of a serial mechanism.

C. Vision-Based Kinematics of a Hexapod

It has been noticed [1] that the lines of the inverse differential kinematic model are the Plücker coordinates of the legs. Indeed, they are the wrenches [46] applied to the moving platform by the actuated legs. Under both interpretations, their expression depends on the chosen reference point. The advantage of taking this point on the mobile platform (for instance, $\mathbf{E}$) is that in such a case, $\overrightarrow{\mathbf{EB}}_i,\ i = 1, \ldots, 6$ are constant, and $\mathbf{D}_e^{\text{inv}}$ only depends on $\underline{\mathbf{u}}_i,\ i = 1, \ldots, 6$. Consequently, if one can measure or estimate (i.e., through a model) $\underline{\mathbf{u}}_i,\ i = 1, \ldots, 6$ in the end-effector frame, one can easily convert ${}^e\tau_e$, the end-effector Cartesian velocity *expressed in the end-effector frame* into joint velocities.

This measure can be done with a camera embedded onto the end-effector (i.e., in a first approximation $\mathcal{R}_c = \mathcal{R}_e$) and observing the legs (see Section III-B). In this case, the vision-based kinematics of the hexapod expressed in the end-effector frame are very simple

$$q_i {}^e\underline{\mathbf{u}}_i = {}^e\mathbf{B}_i - {}^e\mathbf{R}_b {}^b\mathbf{A}_i - {}^e\mathbf{t}_b \tag{6}$$

$$\dot{\mathbf{q}} = {}^e\mathbf{D}_e^{\text{inv}} {}^e\tau_e \tag{7}$$

with

$${}^e\mathbf{D}_e^{\text{inv}} = \begin{pmatrix} {}^e\underline{\mathbf{u}}_1^T & {}^e h_1 {}^e\underline{\mathbf{h}}_1^T \\ \vdots & \vdots \\ {}^e\underline{\mathbf{u}}_6^T & {}^e h_6 {}^e\underline{\mathbf{h}}_6^T \end{pmatrix} \tag{8}$$

$${}^e h_i \underline{\mathbf{h}}_i^e = {}^e\mathbf{B}_i \times {}^e\underline{\mathbf{u}}_i, \quad i = 1, \ldots, 6. \tag{9}$$

This formulation can apply to a classical visual servoing scheme with an embedded camera. Indeed, such a scheme generates, without loss of generality, a desired ${}^e\tau_e$ from the images of an externally fixed target. In practice, it may be awkward, since the camera should observe both the external target and all the legs. Alternately, several cameras could be used, but need be synchronized and calibrated with respect to each other.

In practice, it may be more convenient if the camera observing the legs is fixed to the base. Then, the reference frame associated to it is, without loss of generality, the base frame, and the kinematics of the hexapod do not express as simply as in the end-effector embedded-camera case. Indeed, expressed in the base frame, (5) becomes

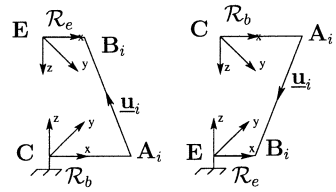


Fig. 4. Duality between the mobile end-effector mode and the fixed end-effector mode.

$${}^b\mathbf{D}_e^{\text{inv}} = \begin{pmatrix} {}^b\underline{\mathbf{u}}_1{}^T & ({}^b\overrightarrow{\mathbf{CB}}_1 \times {}^b\underline{\mathbf{u}}_1)^T \\ \vdots & \vdots \\ {}^b\underline{\mathbf{u}}_6{}^T & ({}^b\overrightarrow{\mathbf{CB}}_6 \times {}^b\underline{\mathbf{u}}_6)^T \end{pmatrix} \tag{10}$$

where ${}^b\overrightarrow{\mathbf{C}^b\mathbf{B}}_i = {}^b\mathbf{R}_e {}^e\mathbf{B}_i\ \forall i = 1, \ldots, 6$. Hence, it is necessary with this expression to estimate the end-effector orientation with respect to the base frame.

An alternate formulation is possible, which is somewhat less useful for standard Cartesian control. However, it is well suited to the observation of the legs only, and thereby to the control scheme proposed in Section IV. It consists of considering the mechanism in its dual operating mode: the end-effector is fixed, and the base moves with respect to it. Thus, we are interested in the inverse differential kinematic model relating the base Cartesian velocity

$${}^b\tau_b = \begin{pmatrix} {}^bV_b \\ {}^b\Omega_b \end{pmatrix}$$

expressed in the base frame to the joint velocities. By analogy with (6)–(9), i.e., by permutation of the roles of $\mathbf{B}_i$ and $\mathbf{A}_i$, and of $\mathcal{R}_e$ and $\mathcal{R}_b$ (Fig. 4), one obtains the vision-based kinematics of the hexapod expressed in the base frame

$$q_i {}^b\underline{\mathbf{u}}_i = {}^b\mathbf{R}_e {}^e\mathbf{B}_i + {}^b\mathbf{t}_e - {}^b\mathbf{A}_i \tag{11}$$

$$\dot{q} = {}^b\mathbf{D}_b^{\text{inv}} {}^b\tau_b \tag{12}$$

with

$${}^b\mathbf{D}_b^{\text{inv}} = -\begin{pmatrix} {}^b\underline{\mathbf{u}}_1{}^T & {}^b h_1 {}^b\underline{\mathbf{h}}_1^T \\ \vdots & \vdots \\ {}^b\underline{\mathbf{u}}_6{}^T & {}^b h_6 {}^b\underline{\mathbf{h}}_6^T \end{pmatrix} \tag{13}$$

$${}^b h_i {}^b\underline{\mathbf{h}}_i = {}^b\mathbf{A}_i \times {}^b\underline{\mathbf{u}}_i = {}^b\mathbf{B}_i \times {}^b\underline{\mathbf{u}}_i, \quad i = 1, \ldots, 6. \tag{14}$$

Notice the minus signs in (11) and (13), coming from the fact that in the permutation the direction of the legs has changed (Fig. 4). Notice also that now the inverse differential kinematic model is independent from the relative pose of the end-effector and base, since it only depends on the measured ${}^b\underline{\mathbf{u}}_i$'s and the constant kinematic parameters ${}^b\mathbf{A}_i$'s.

Notice also that (14) uses the fact that $h\underline{\mathbf{h}}$ is the same for any point on the line.

D. Vision-Based Kinematics Expressed in the Camera Frame

In practice, the camera is not co-located with the base frame of the robot, nor share the same orientation in space. Consequently, the camera-to-base transformation ${}^c\mathbf{T}_b$ differs from the identity, and the above model (11)–(13) may be questioned.

Nevertheless, it is easy to show that one only needs to replace b by c in the super- and subscripts to convert the vision-based kinematics expressed in the base frame into vision-based kinematics expressed in the camera frame.

III. Discussion

Here, the proposed model is discussed, with explicit regard to the sensing problem and to control. Indeed, no matter the control law, it will make use of some differential kinematic model, and thus will need to measure or estimate the $\underline{\mathbf{u}}_i$'s. Moreover, the derivation of (11)–(13) does not absolutely imply that computer vision should be used for that.

A. Why Vision Should Be Used Rather Than Joint Sensors

There are three manners to measure or estimate the $\underline{\mathbf{u}}_i$'s. The first one, which, of course, we discard immediately, is to estimate the end-effector pose with respect to the base by numerical inversion of the inverse kinematic model, and then use (11) to obtain $\underline{\mathbf{u}}_i$.

The second manner is to place joint sensors in the $\mathbf{A}_i$'s, so that they would deliver the pointing direction of the leg. This manner is valid, since the $\underline{\mathbf{u}}_i$'s would be measured and not estimated through a delicate numerical procedure, as above. Nevertheless, it is, in our opinion, still not the correct manner. Indeed, a joint sensor delivers a value expressed *in its own* reference frame. To convert this value in the base frame would require either an extremely accurate assembly procedure, or the identification of the relative orientation of *each* joint sensor frame with respect to the base frame. Moreover, additional joint offsets would need be identified.

Since the leading vector of a leg is essentially a Cartesian feature, we chose to estimate it by vision. Indeed, vision is an adequate tool for Cartesian sensing, and, following [26], if vision is also chosen for calibration, this does not add an extra calibration parameter. It will even be shown in Section V that using vision reduces the parameter set needed for control.

B. Cylindrical Leg Observation

Now the problem is to recover ${}^c\underline{\mathbf{u}}_i$ from the leg observation. It may be somehow tedious, although certainly feasible, in the case of an arbitrary shape. Hopefully, for mechanical reasons such as rigidity, most of the parallel mechanisms are not only designed with slim and rectilinear legs, but, even better, with cylindrical shapes.

Except in the degenerated case where the projection center lies on the cylinder axis, the visual edge of a cylinder is a straight line (Fig. 5). Consequently, it can be represented by its binormalized Plücker coordinates in the camera frame. Let us note ${}^c\underline{\mathbf{h}}^{e_1}$ and ${}^c\underline{\mathbf{h}}^{e_2}$ as the image projections of the two edges of a cylinder. These two vectors are oriented so that they point from the cylinder revolution axis outwards. Then, their expression is related to the binormalized Plücker coordinates $({}^c\underline{\mathbf{u}}_i, {}^c\underline{\mathbf{h}}_i, {}^ch_i)$ of the cylinder axis (see Fig. 6) by

$$ {}^c\underline{\mathbf{h}}^{e_1} = -\cos\theta\, {}^c\underline{\mathbf{h}} - \sin\theta\, {}^c\underline{\mathbf{u}} \times {}^c\underline{\mathbf{h}} \tag{15} $$

$$ {}^c\underline{\mathbf{h}}^{e_2} = +\cos\theta\, {}^c\underline{\mathbf{h}} - \sin\theta\, {}^c\underline{\mathbf{u}} \times {}^c\underline{\mathbf{h}} \tag{16} $$

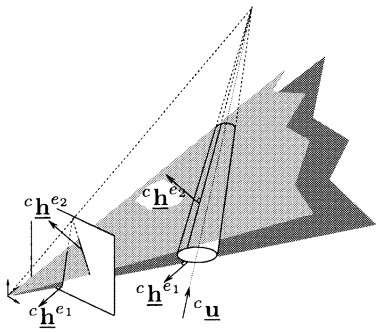


Fig. 5. Projection of a cylinder in the image.

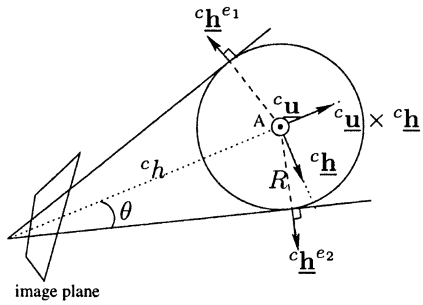


Fig. 6. Construction of the visual edges of a cylinder. The cylinder is viewed from the top.

where $\cos\theta = \sqrt{{}^ch^2 - R^2}/{}^ch$ and $\sin\theta = R/{}^ch$, and R is the cylinder radius.

From (15) and (16), it is easy to show that the leading vector ${}^c\underline{\mathbf{u}}$ of the cylinder axis, expressed in the camera frame, writes

$$ {}^c\underline{\mathbf{u}} = \frac{{}^c\underline{\mathbf{h}}^{e_1} \times {}^c\underline{\mathbf{h}}^{e_2}}{\| {}^c\underline{\mathbf{h}}^{e_1} \times {}^c\underline{\mathbf{h}}^{e_2} \|}. \tag{17} $$

Notice that the geometric interpretation of this result is that ${}^c\underline{\mathbf{u}}$ is, up to a scale factor, the vanishing point of the two image edges, i.e., their intersection point in the image.

IV. Vision-Based Control

In this section, the control problem is addressed: from a given configuration of the hexapod legs observed by a camera attached to the base, how to reach a desired configuration?

Visual servoing is based on the so-called interaction matrix $\mathbf{L}_{(s)}^T$ [47] which relates the instantaneous relative motion $T_c = {}^c\tau_c - {}^c\tau_s$ between the camera and the scene, to the time derivative of the vector s of all the visual primitives that are used through

$$ \dot{s} = \mathbf{L}_{(s)}^T T_c \tag{18} $$

where ${}^c\tau_c$ and ${}^c\tau_s$ are, respectively, the kinematic *twist*[2] of the camera and the scene, both expressed in $\mathcal{R}_c$.

Then, one achieves exponential decay of an error $e(s, s_d)$ between the current primitive vector s and the desired one s_d using

[2]Note that there is a usual mistake in most of the papers dealing with visual servoing between screws and twists, refer, for instance, to [46] and [48].

a proportional linearizing and decoupling control scheme of the form

$$T_c = -\lambda \hat{\mathbf{L}}_{(s)}^{T+} e(s, s_d) \tag{19}$$

where T_c is used as a pseudocontrol variable.

In this paper, we will similarly need to define a visual primitive, then form an error between its current value and its desired one, then relate in some way its time derivative to the actuation, and finally find a control relation between the error and the actuation.

A. Visual Primitive and Error

As foreseen above, the unit vectors ${}^c\underline{\mathbf{u}}_i,\ i = 1, \ldots, 6$ will be used as visual primitives. Since these primitives are expressed in the 3-D space, the control scheme is close to a PBVS scheme. However, since the reconstruction step (17) is algebraic, it is, nevertheless, not far away from IBVS.

The visual primitives being unit vectors, it is theoretically more elegant to use the geodesic error rather than the standard vector difference. Consequently, the error grounding the proposed control law will be

$$\mathbf{e}_i = {}^c\underline{\mathbf{u}}_i \times {}^c\underline{\mathbf{u}}_{di}. \tag{20}$$

B. Interaction Matrix

Here, the time derivative of ${}^c\underline{\mathbf{u}}_i$ is related to the actuation. From (3), one immediately obtains

$${}^c\dot{\underline{\mathbf{u}}}_i = \frac{1}{q_i}\frac{d}{dt}{}^c\mathbf{A}_i \overrightarrow{{}^c\mathbf{B}_i} - \frac{\dot{q}_i}{q_i}{}^c\underline{\mathbf{u}}_i. \tag{21}$$

Inserting the interaction matrix associated with a 3-D point [19] applied to the moving point ${}^c\mathbf{B}_i$

$$\frac{d}{dt}{}^c\mathbf{A}_i \overrightarrow{{}^c\mathbf{B}_i} = \begin{bmatrix} -\mathbf{I}_3 & [{}^c\mathbf{B}_i]_\times \end{bmatrix} {}^c\tau_c \tag{22}$$

where $[\,]_\times$ is the antisymmetric matrix associated with the cross product into (21), yields

$${}^c\dot{\underline{\mathbf{u}}}_i = -\frac{1}{q_i}\begin{bmatrix} \mathbf{I}_3 & -[{}^c\mathbf{B}_{i\times}] \end{bmatrix} {}^c\tau_c - \frac{\dot{q}_i}{q_i}{}^c\underline{\mathbf{u}}_i. \tag{23}$$

It is interesting to see that both the base Cartesian velocity and the joint velocity vector appear in this expression, while also being linked to each other by the inverse differential kinematic model in (13). This is certainly due to the existence of closed kinematic chains.

Nevertheless, using precisely the linking inverse differential kinematic model, one can exhibit a relationship between each ${}^c\dot{\underline{\mathbf{u}}}_i$ and ${}^c\tau_c$ only. Indeed, each line of the inverse differential kinematic model in (13) rewrites as

$$-\begin{bmatrix} {}^c\underline{\mathbf{u}}_i^T & {}^ch_1{}^c\underline{\mathbf{h}}_i^T \end{bmatrix} = -{}^c\underline{\mathbf{u}}_i^T \begin{bmatrix} \mathbf{I}_3 & -[{}^c\mathbf{B}_i]_\times \end{bmatrix}. \tag{24}$$

Hence, one gets the following relationship:

$${}^c\dot{\underline{\mathbf{u}}}_i = \mathbf{M}_i^T {}^c\tau_c \tag{25}$$

$$\mathbf{M}_i^T = -\frac{1}{q_i}\left(\mathbf{I}_3 - {}^c\underline{\mathbf{u}}_i {}^c\underline{\mathbf{u}}_i^T\right)\begin{bmatrix} \mathbf{I}_3 & -[{}^c\mathbf{B}_i]_\times \end{bmatrix} \tag{26}$$

where $\mathbf{M}_i^T$ is 3×6 and is obviously of rank 2.

Since ${}^c\mathbf{B}_i = {}^c\mathbf{A}_i + q_i{}^c\underline{\mathbf{u}}_i$, one can rewrite the above expression, using uniquely constant (${}^c\mathbf{A}_i$) or easily measurable (q_i and ${}^c\underline{\mathbf{u}}_i$) quantities, as

$$\mathbf{M}_i^T = -\frac{1}{q_i}\left(\mathbf{I}_3 - {}^c\underline{\mathbf{u}}_i {}^c\underline{\mathbf{u}}_i^T\right)\begin{bmatrix} \mathbf{I}_3 & -[{}^c\mathbf{A}_i + q_i{}^c\underline{\mathbf{u}}_i]_\times \end{bmatrix}. \tag{27}$$

Necessary Condition 1: A minimum of three independent legs is necessary to control the end-effector pose, provided that there exists a diffeomorphism between the task space and the end-effector pose (i.e., provided that the observation of three legs allows determining the end-effector pose).

However, it should be borne in mind that this is solely a necessary condition, but that it does not hold any sufficient counterpart. The implication of this necessary condition is that the proposed approach might be used for parallel mechanisms similar to the Gough platform, but with less than six DOFs.

A global *pseudointeraction matrix* $\mathbf{M}^T$ can then be obtained by stacking each individual interaction matrices $\mathbf{M}_i^T,\ i = 1, \ldots, 6$. However, it is, in our opinion, an open question whether $\mathbf{M}$ shall or shall not be considered as an interaction matrix. Indeed, in visual servoing, the various visual primitives are the image projections of objects in space that are rigidly linked to each other, while here, each of the legs is in relative motion with respect to the other ones. Nevertheless, effective control can be derived as shown in the following section.

C. Control Law

Let us choose a control such that $E = (\mathbf{e}_1^T, \ldots, \mathbf{e}_6^T)^T$ decreases exponentially, i.e., such that

$$\dot{E} = -\lambda E. \tag{28}$$

Then, introducing $\mathbf{N}_i^T = -[{}^c\underline{\mathbf{u}}_{di}]_\times \mathbf{M}_i^T$ and $\mathbf{N}^T = (\mathbf{N}_1, \ldots, \mathbf{N}_6)^T$, the combination of (20), (25), and (28) gives

$$\dot{E} = \mathbf{N}^T {}^c\tau_c \tag{29}$$

which yields

$$\mathbf{N}^T {}^c\tau_c = -\lambda E. \tag{30}$$

The Cartesian pseudocontrol velocity is, hence

$${}^c\tau_c = -\lambda \widehat{\mathbf{N}^T}^+ E \tag{31}$$

and can be transformed into the control joint velocities using (12)

$$\dot{q} = -\lambda {}^c\widehat{\mathbf{D}_c^{\text{inv}}}\widehat{\mathbf{N}^T}^+ E \tag{32}$$

where the *hat* means that only an estimate can be used.

Notice that since the joint values appear marginally in (27), a control scheme which *does not make any use of the joint values* can be set up by using the median joint values $q_{i_{\text{med}}} = (q_{i_{\max}} + q_{i_{\min}})/2$ [49].

D. Stability

Let us consider the error function $E' = {}^c\widehat{\mathbf{D}}_c^{\text{inv}}\widehat{\mathbf{N}^T}^+ E$. Then, the proposed control rewrites as $\dot{\mathbf{q}} = -\lambda E'$.

Let us now consider the time derivative of E' under the latter control. It can easily be shown, from (29) and inverting (12), that

$$\dot{E}' = -\lambda {}^c\widehat{\mathbf{D}}_c^{\text{inv}}\widehat{\mathbf{N}^T}^+ \mathbf{N}^T \, {}^c\mathbf{D}_c^{\text{inv}+} E' \tag{33}$$

where ${}^c\mathbf{D}_c^{\text{inv}+}\dot{\mathbf{q}}$ encodes the *actual* motion of the end-effector under the effect of the joint actuation and all unmodeled phenomena.

As a consequence, an evident exponential stability condition of the error function E' is

$${}^c\widehat{\mathbf{D}}_c^{\text{inv}}\widehat{\mathbf{N}^T}^+ \mathbf{N}^T {}^c\mathbf{D}_c^{\text{inv}+} > 0. \tag{34}$$

This condition is ensured if the following two conditions are satisfied:

$${}^c\widehat{\mathbf{D}}_c^{\text{inv}} {}^c\mathbf{D}_c^{\text{inv}+} > 0 \tag{35}$$

$$\widehat{\mathbf{N}^T}^+ \mathbf{N}^T > 0. \tag{36}$$

In turn, these conditions are satisfied if the measured ${}^c\underline{\mathbf{u}}_i$'s are sufficiently close to the real ones, and if the ${}^c\mathbf{A}_i$'s are accurately enough calibrated and if no singular configuration occurs. The singular configurations can be of two kinds: the much-studied kinematic singularities of the robot itself, or configurations such that $\mathbf{N}^T$ is rank-deficient (that are left to future investigation).

V. Vision-Based Calibration

As seen above, the only kinematic parameters the control law depends on are the attachment points of the legs onto the base expressed in the *camera* frame $({}^c\mathbf{A}_i)$ and the joint offsets. The latter appear in two places in (27): under the form ${}^c\mathbf{A}_i + q_i {}^c\underline{\mathbf{u}}_i$ and as a gain. Considering the order of magnitude of ${}^c\mathbf{A}_i$ and q_i, one can neglect small errors on the joint offsets. Moreover, since the joints are prismatic, it is easy to measure their offsets manually with a millimetric accuracy. This is also highly sufficient to ensure that the gain is accurate enough. This means that, as far as control is concerned, there is no need for identifying the other usual kinematic parameters: attachment points onto the mobile platform and joint offsets.

In [26], a calibration procedure was proposed, using legs observation, where in a first step, the points $\mathbf{A}_i$ were estimated in the camera frame, then in a second step, were expressed in the base frame, and finally, the others kinematic parameters where deduced. Essentially, only the first step of this procedure is needed.

This step is reformulated here in a more elegant way, using the binormalized Plücker coordinates of the cylinder edges (15), and minimizing an image-based criterion.

Assuming that the attachment point $\mathbf{A}_i$ is lying on the revolution axis of the leg and referring again to Fig. 6, one obtains, for any leg i and robot configuration j

$${}^c\underline{\mathbf{h}}_{i,j}^{e_1\,T} {}^c\mathbf{A}_i = -R \tag{37}$$

$${}^c\underline{\mathbf{h}}_{i,j}^{e_2\,T} {}^c\mathbf{A}_i = -R. \tag{38}$$

Stacking such relations for n_c robot configurations yields the following linear system:

$$\begin{pmatrix} {}^c\underline{\mathbf{h}}_{i,1}^{e_1\,T} \\ {}^c\underline{\mathbf{h}}_{i,1}^{e_2\,T} \\ \vdots \\ {}^c\underline{\mathbf{h}}_{i,n_c}^{e_1\,T} \\ {}^c\underline{\mathbf{h}}_{i,n_c}^{e_2\,T} \end{pmatrix} {}^c\mathbf{A}_i = \begin{pmatrix} -R \\ -R \\ \vdots \\ -R \\ -R \end{pmatrix} \tag{39}$$

which has a unique least-square solution if there are at least two configurations with different leg directions.

The calibration procedure is, hence, reduced to a strict minimum. To improve its numerical efficiency, one should only take care to use robot configurations with the larger angles between each leg direction. However, since the ${}^c\mathbf{A}_i$ only appear in the pseudointeraction matrix, they do not require a very accurate estimation.

One could argue that, in practice, using a CAD model might be largely sufficient, except that a CAD model is not expressed in the camera frame, but in the base frame. Thus, one either has to estimate the base-to-camera transformation (which is a nonlinear problem), or to simply run our linear calibration scheme.

VI. Results

The commercial DeltaLab *Table de Stewart* in Fig. 3 was simulated and controlled experimentally. It is such that for all $k \in \{0,1,2\}$

$${}^b\mathbf{A}_{2k} = R_b \begin{pmatrix} \cos\left(k\frac{\pi}{3} + \alpha\right) \\ \sin\left(k\frac{\pi}{3} + \alpha\right) \\ 0 \end{pmatrix}$$

$${}^b\mathbf{A}_{2k+1} = R_b \begin{pmatrix} \cos\left(k\frac{\pi}{3} - \alpha\right) \\ \sin\left(k\frac{\pi}{3} - \alpha\right) \\ 0 \end{pmatrix}$$

$${}^e\mathbf{B}_{2k} = R_e \begin{pmatrix} \cos\left(k\frac{\pi}{3} + \beta\right) \\ \sin\left(k\frac{\pi}{3} + \beta\right) \\ 0 \end{pmatrix}$$

$${}^e\mathbf{B}_{2k+1} = R_e \begin{pmatrix} \cos\left(k\frac{\pi}{3} - \beta\right) \\ \sin\left(k\frac{\pi}{3} - \beta\right) \\ 0 \end{pmatrix}$$

with $R_b = 270$ mm, $\alpha = 4.25°$, $R_e = 195$ mm, $\beta = 5.885°$, and the legs range are [345 mm, 485 mm].

A. Image Noise

It is not immediate to model realistically the effect of image noise on detecting the visual edges of a cylinder. One can imagine to use the (ρ, θ) representation of an image line and to add noise to these two components. However, it is not certain that the two noise components are uncorrelated. Alternately, one can also determine the virtual intersections of the visual edge with the image border and to add noise to these intersections [50].

An alternative model is presented here, which takes advantage of the fact that an image line is essentially a unit vector. Thus, image noise will necessary result into a rotation of this unit vector. Consequently, one needs to define a random rotation matrix, that is to say, a rotation axis and a positive rotation angle. Preliminary tests showed that simply taking this axis as a

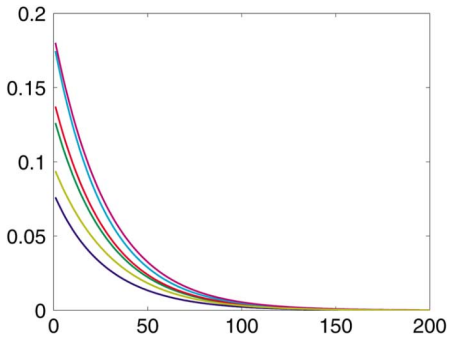


Fig. 7. Errors (unit less) on each leg $\mathbf{e}_i^T \mathbf{e}_i$ with respect to time (expressed as iteration number).

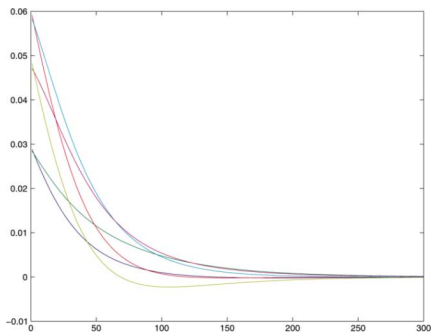


Fig. 8. Joint velocities (m/s) with respect to time.

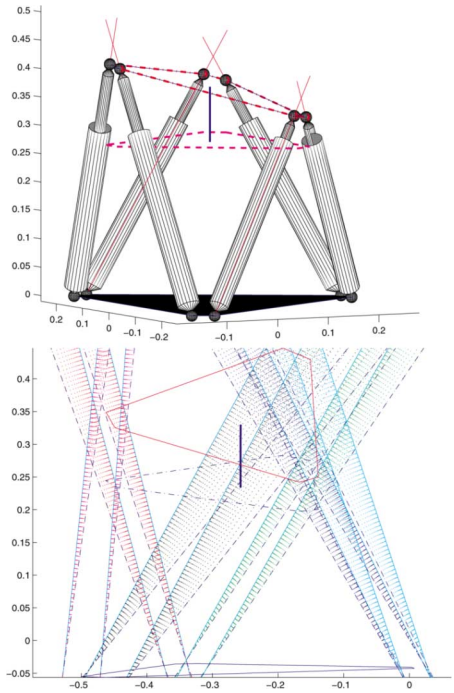


Fig. 9. Top: Trajectory in space with initial (magenta, dashed) and desired (red, dash-dotted) position of the platform. Bottom: Image trajectories.

uniformly distributed unit vector and this angle as a uniformly distributed positive scalar gives a realistic behavior.

This noise model has the advantages that it is easy to implement, it does not depend on the simulated image size, and is parametered by a single scalar (i.e., the maximal amplitude of the rotation angle).

To give an idea of how to choose this maximal amplitude, an error of about ± 1 pixel on the extremities of a 300 pixel-long line segment yields a rotation angle of approximately 0.05°.

B. Control Validation

In all the simulations presented here and below, the initial configuration of the platform is the reference configuration where all the legs have minimal length. The goal configuration is obtained from this reference configuration by a translation by 10 cm along the z axis of the platform (upward vertical) and a rotation of 15° around the x axis, thus reaching the workspace limit.

The control law proposed in this paper was first tested without any noise for validation and analysis purposes.

In a first simulation, all the legs are used for control. Fig. 7 shows that the errors on each leg converge exponentially to zero, with a perfect decoupling in the task space. Moreover, the joint velocities (Fig. 8) have a smooth behaviour. Finally, Fig. 9 shows that the desired end-effector pose is reached and the trajectory of the legs in the image.

C. Calibration Validation

In order to quantify the calibration procedure, the simulated robot was calibrated by moving it into its 64 extremal configurations (i.e., each leg joint is extended to its lower limit, then to its upper limit). In each configuration, the visual edges of each leg were generated from the inverse kinematic model and added noise as described above, with a maximal amplitude of 0.05°. This calibration procedure was repeated 100 times. As a result, the median error on each attachment point is less than 1 mm. Recall that this result is obtained linearly without making any prior assumption on the mechanism geometry, nor on the camera-base transformation.

D. Realistic Simulation

Now, a more realistic simulation is presented, where the calibration is first performed using the 64 extremal configurations, then control is launched using the calibrated values. Noise is added as above during image detection, with amplitudes of 0.01°, 0.05°, and 0.1°.

It is noticeable that the calibration errors (in terms of maximal error on each of the component of each attachment point) is, respectively, of 0.5, 1.4, and 10 mm. Additionally, the median error of the convergence tails are, respectively, 0.1, 0.6, and 1.1 mm, while the maximal error are 0.6, 1.9, and 3 mm. Graphically (Fig. 10), the sum of the errors on each leg $E^T E$ still decreases, yielding a potentially good robustness.

E. Experiments

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 approximate 20 ms sampling period (this part is not yet running under RTAI, but only under standard Linux). Frame grabbing, line tracking, and numerical computation are performed using ViSP, an open C++ library for visual servoing [51]. The camera used was a 15 Hz, 1024×768, IEEE1394 camera with 4.8 mm focal length placed approximately 1 m from the base center (see Fig. 3).

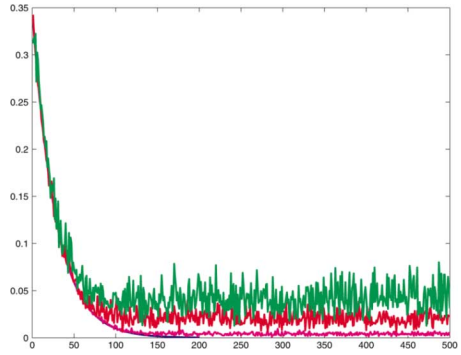


Fig. 10. Robustness to noise: sum of squares of the errors $E^T E$ with a noise amplitude of 0°, 0.01°, 0.05°, and 0.1°.

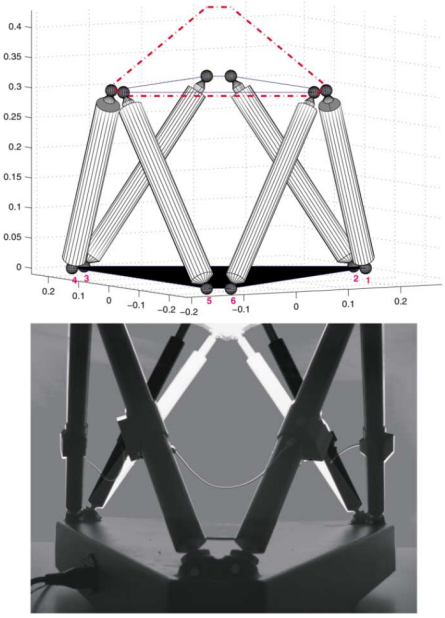


Fig. 11. Composition of the desired (dashed platform in the CAD view, black in the camera image) and initial (solid platform in the CAD view, white in the camera image) configurations.

It also has to be noticed that the mechanism presents high frictious disturbances that have not yet been compensated for, since friction seems to depend nontrivially 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 platform was asked to move from a configuration where all the legs are retracted to one where the two rear legs are stretched out (see Fig. 11). The resulting evolution of the controlled leg direction with respect to time is displayed in Fig. 12, showing the convergence of the control despite a high noise level.

VII. Conclusion

A novel approach was proposed for controlling a parallel mechanism using vision as a redundant sensor, adding a proprioceptive nature to the usual exteroceptive nature of vision. Within this approach, the standard tryptic "modeling, identification, and control" is reformulated, yielding a *unified vision-based* modeling, identification, and control framework. It is not only theoretically extremely elegant, but appears to own interesting properties. Indeed, stability conditions were exhibited, as well as potentially good robustness properties. It was validated and illustrated on a Gough–Stewart platform. Nevertheless, the results presented here should apply for any robot of the same kind with at least three length-actuated legs linked with passive revolute, universal, or spherical joints to the base and end-effector.

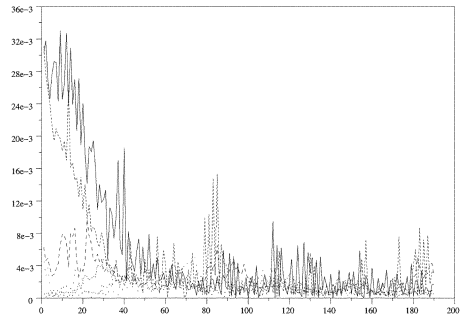


Fig. 12. Evolution of the controlled leg direction errors with respect to time.

However, this paper is only the seed of a vast research domain. Indeed, there are several points to be addressed before a safe and satisfying implementation can be made on a real platform. First, this control does not take into account joint limit avoidance. This point is fundamental, since these limits can easily be reached and their avoidance may not be as trivial as for serial mechanisms. Second, this control assumes a detection of cylinder edges, which is known to be delicate in vision. Moreover, as can be seen in Fig. 11, using a single camera may yield visual self-occlusions of the hexapod. To avoid such occlusions, one may use several cameras, but this would require additional calibration, intercamera synchronization, and collaboration. Alternately, we plan to use a panoramic camera placed between the mechanism legs.

Third, since the control is essentially based on the direction of each leg, one may think of extracting the latter from the image of a generally shaped leg. Finally, we plan to extend this framework to any parallel mechanism, and we started to put effort into the development of high-speed vision tools to cope with the high velocities of parallel mechanisms [52].

References

[1] J.-P. Merlet, *Parallel Robots*. Norwell, MA: Kluwer, 2000.

[2] J. Wang and O. Masory, "On the accuracy of a Stewart platform—Part I: The effect of manufacturing tolerances," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1993, pp. 114–120.

[3] [Online]. Available: http://www-sop.inria.fr/coprin/equipe/merlet.

[4] J.-P. Merlet, An algorithm for the forward kinematics of general 6 D.O.F. parallel manipulators INRIA, Sophia Antipolis, France, Tech. Rep. 1331, Nov. 1990.

[5] M. Husty, "An algorithm for solving the direct kinematics of general Gough-Stewart platforms," *Mech. Mach. Theory*, vol. 31, no. 4, pp. 365–380, 1996.

[6] P. Dietmaier, "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. Norwell, MA: Kluwer, 1998, pp. 1–10.

[7] V. Gough and S. Whitehall, "Universal tyre test machine," in *Proc. FISITA 9th Int. Tech. Congr.*, May 1962, pp. 117–137.

[8] D. Stewart, "A platform with six degrees of freedom," in *Proc. IMechE (London)*, 1965, vol. 180, no. 15, pp. 371–386.

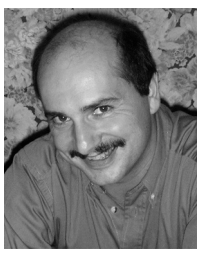
[9] X. Zhao and S. Peng, "Direct displacement analysis of parallel manipulators," *J. Robot. Syst.*, vol. 17, no. 6, pp. 341–345, 2000.

[10] H. Kim and L.-W. Tsui, "Evaluation of a Cartesian parallel manipulator," in *Advances in Robot Kinematics: Theory and Applications*, J. Lenarčič and F. Thomas, Eds. Norwell, MA: Kluwer, Jun. 2002.

[11] G. Gogu, "Fully-isotropic T3Rl-type parallel manipulator," in *On Advances in Robot Kinematics*, J. Lenarčič and C. Galletti, Eds. Norwell, MA: Kluwer, 2004, pp. 265–272.
[12] L. Baron and J. Angeles, "The on-line direct kinematics of parallel manipulators under joint-sensor redundancy," in *Advances in Robot Kinematics*. Norwell, MA: Kluwer, 1998, pp. 126–137.
[13] F. Marquet, "Contribution à l'étude de l'apport de la redondance en robotique parallele," Ph.D. dissertation, LIRMM-Univ. Montpellier II, Montpellier, France, Oct. 2002.
[14] D. DeMenthon and L. Davis, "Model-based object pose in 25 lines of code," *Int. J. Comput. Vis.*, vol. 15, no. 1/2, pp. 123–141, Jun. 1995.
[15] J. Lavest, M. Viala, and M. Dhome, "Do we really need an accurate calibration pattern to achieve a reliable camera calibration," in *Proc. ECCV*, Freiburg, Germany, Jun. 1998, pp. 158–174.
[16] N. Andreff, P. Renaud, P. Martinet, and F. Pierrot, "Vision-based kinematic calibration of an H4 parallel mechanism: Practical accuracies," *Ind. Robot: An Int. J.*, vol. 31, no. 3, pp. 273–283, May 2004.
[17] B. Dasgupta and T. Mruthyunjaya, "Force redundancy in parallel manipulators: Theoretical and practical issues," *Mech. Mach. Theory*, vol. 33, no. 6, pp. 727–742, 1998.
[18] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
[19] P. Martinet, J. Gallice, and D. Khadraoui, "Vision based control law using 3D visual features," in *Proc. World Autom. Congr., Robot. Manuf. Syst.*, Montpellier, France, May 1996, vol. 3, pp. 497–502.
[20] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 525–532, Aug. 1998.
[21] H. Christensen and P. Corke, Eds., *Int. J. Robot. Res. Special Issue Visual Servoing*, vol. 22, no. 10/11, Oct. 2003.
[22] M. Koreichi, S. Babaci, F. Chaumette, G. Fried, and J. Pontnau, "Visual servo control of a parallel manipulator for assembly tasks," in *Proc. 6th Int. Symp. Intell. Robot. Syst.*, Edinburgh, U.K., Jul. 1998, pp. 109–116.
[23] H. Kino, C. Cheah, S. Yabe, S. Kawamua, and S. Arimoto, "A motion control scheme in task oriented coordinates and its robustness for parallel wire driven systems," in *Proc. Int. Conf. Adv. Robot.*, Tokyo, Japan, Oct. 25–27, 1999, pp. 545–550.
[24] P. Kallio, Q. Zhou, and H. N. Koivo, "Three-dimensional position control of a parallel micromanipulator using visual servoing," in *Proc. SPIE Microrobot. Microassembly II*, B. J. Nelson and E. J. M. Breguet, Eds., Boston, MA, Nov. 2000, vol. 4194, pp. 103–111.
[25] P. Renaud, N. Andreff, P. Martinet, and G. Gogu, "Kinematic calibration of parallel mechanisms: A novel approach using legs observation," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 529–538, Aug. 2005.
[26] P. Renaud, N. Andreff, G. Gogu, and P. Martinet, "On vision-based kinematic calibration of a Stewart-Gough platform," in *Proc. 11th World Congr. Mech. Mach. Sci.*, Tianjin, China, Apr. 1–4, 2004, pp. 1906–1911.
[27] P. Renaud, N. Andreff, F. Pierrot, and P. Martinet, "Combining end-effector and legs observation for kinematic calibration of parallel mechanisms," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, May 2004, pp. 4116–4121.
[28] J. Plücker, "On a new geometry of space," *Philos. Trans. Roy. Soc. London*, vol. 155, pp. 725–791, 1865.
[29] J. Semple and G. Kneebone, *Algebraic Projective Geometry*. Oxford, U.K.: Oxford Science, 1952.
[30] J. M. MacCarthy, *Introduction to Theoretical Kinematics*. Cambridge, MA: MIT Press, 1990.
[31] H. Pottmann, M. Peternell, and B. Ravani, "Approximation in line space—Applications in robot kinematics and surface reconstruction," in *Advances in Robot Kinematics: Analysis and Control*, J. Lenarčič and M. L. Husty, Eds. Norwell, MA: Kluwer, 1998, pp. 403–412.
[32] O. Faugeras, *Three-Dimensional Computer Vision—A Geometric Viewpoint*, ser. Artif. Intell.. Cambridge, MA: MIT Press, 1993.
[33] R. Mahony and T. Hamel, "Visual servoing using linear features for under-actuated rigid-body dynamics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, 2001, pp. 1153–1158.
[34] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 679–700, Aug. 2002.
[35] R. Mahony and T. Hamel, "Image-based visual servo control of aerial robotic systems using linear image features," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 227–239, Apr. 2005.
[36] N. Andreff, A. Marchadier, and P. Martinet, "Vision-based control of a Gough-Stewart parallel mechanism using legs observation," in *Proc. Int. Conf. Robot. Autom.*, Barcelona, Spain, May 2005, pp. 2546–2551.
[37] P. Renaud, N. Andreff, G. Gogu, and P. Martinet, "On vision-based kinematic calibration of n legs parallel mechanisms," in *Proc. 13th IFAC Symp. Syst. Identification*, Rotterdam, The Netherlands, Aug. 27–29, 2003, pp. 977–982.
[38] [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_jdoc/.
[39] D. Brown, "Close-range camera calibration," *Photogram. Eng.*, vol. 8, no. 37, pp. 855–866, 1971.
[40] R. Tsai, "An efficient and accurate calibration technique for 3D machine vision," in *Proc. Int. Conf. Comput. Vis. Pattern Recog.*, Miami, FL, 1986, pp. 364–374.
[41] O. Faugeras and G. Toscani, "Camera calibration for 3D computer vision," in *Proc. Int. Workshop Mach. Vis. Mach. Intell.*, Tokyo, Japan, 1987, pp. 240–247.
[42] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Mach. Vis. Applic.*, vol. 13, pp. 14–24, 2001.
[43] E. Malis, J. Borrelly, and P. Rives, "Intrinsics-free visual servoing with respect to straight lines," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Lausanne, Switzerland, Oct. 2002, pp. 384–389.
[44] D. Khadraoui, R. Rouveure, C. Debain, P. Martinet, P. Bonton, and J. Gallice, "Vision based control in driving assistance of agricultural vehicles," *Int. J. Robot. Res.*, vol. 17, no. 10, pp. 1040–1054, Oct. 1998.
[45] J. Stolfi, *Oriented Projective Geometry*. New York: Academic, 1991.
[46] J. Angeles, *Fundamentals of Robotic Mechanical Systems*, 2nd ed. New York: Springer-Verlag, 2002.
[47] F. Chaumette, "La commande des robots manipulateurs, Traité IC2" 2002, pp. 105–150, Hermes, ch. Asservissement visuel.
[48] J. Gallardo, J.-M. Rico, A. Frisoli, D. Checcacci, and M. Bergamasco, "Dynamics of parallel manipulators by means of screw theory," *Mech. Mach. Theory*, vol. 38, no. 11, pp. 1113–1131, 2003.
[49] N. Andreff and P. Martinet, "Visual servoing of a Gough-Stewart parallel robot without proprioceptive sensors," in *Proc. 5th Int. Workshop Robot Motion, Control*, Dymaczewo, Poland, Jun. 23–25, 2005, pp. 225–230.
[50] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," *Int. J. Comput. Vis.*, vol. 57, no. 3, pp. 159–178, 2004.
[51] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: A generic software platform with a wide class of robot control skills," *IEEE Robot. Autom. Mag.*, vol. 12, no. 4, pp. 40–52, Dec. 2005.
[52] O. Ait-Aider, N. Andreff, P. Martinet, and J.-M. Lavest, "Simultaneous pose and velocity measurement for high-speed robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, May 2006, pp. 3742–3747.

Nicolas Andreff received the Engineer degree in 1994 from the Ecole Nationale Supérieure d'Electronique, d'Electrotechnique, d'Informatique et d'Hydraulique de Toulouse, Toulouse, France, and the Ph.D. degree in 1999 in computer graphics, computer vision and robotics from the Institut National Polytechnique de Grenoble, Grenoble, France.

Since 2000, he has been an Associate Professor with Institut Français de Mécanique Avance, Clermont-Ferrand, France. His current research interests are in the field of modeling, identification, and control of mechanisms using computer vision.



Philippe Martinet graduated from the Centre Universitaire Scientifique et Technique de Clermont-Ferrand (CUST), Clermont-Ferrand, France, in 1985, and received the Ph.D. degree in electronics science from the Blaise Pascal University, Clermont-Ferrand, France, in 1987.

From 1990 to 2000, he was an Associate Professor at CUST, and since 2000, has been a Professor with the Institut Français de Mécanique Avance, Clermont-Ferrand, France. He is performing research at the Robotics and Vision Group of LASMEA-CNRS, Clermont-Ferrand, France. His research interests include visual servoing, vision-based control, robust control, automatic guided vehicles, active vision and sensor integration, visual tracking, and parallel architecture for visual servoing applications.