Minimal Representation for the Control of Parallel Robots via Leg Observation Considering a Hidden Robot Model

Sébastien Briot, Victor Rosenzveig, Philippe Martinet, Erol Ozgür, Nicolas Bouton

Minimal Representation for the Control of Parallel Robots via Leg Observation Considering a Hidden Robot Model

Sébastien Briot[a,*], Victor Rosenzveig[a], Philippe Martinet[a,b], Erol Özgür[c], Nicolas Bouton[c]

[a]*Institut de Recherches en Communications et Cybernétique de Nantes (IRCCyN)*
UMR CNRS 6597 – 1 rue de la Noë, 44321 Nantes, France
[b]*École Centrale de Nantes, 1 rue de la Noë, 44321 Nantes, France*
[c]*Institut Pascal (UMR 6602) – Sigma Clermont – Campus des Cézeaux, 27 Rue Roche Genès, 63178 Aubière, France*

Abstract

Previous works on the visual servoing of parallel robots using the observation of their leg directions validated the feasibility of the approach but they have enlighten two main surprising results for which no answer was given: (i) the observed robot which is composed of n legs could be controlled in most cases using the observation of only m leg directions ($m < n$), and that (ii) in some cases, the robot did not converge to the desired end-effector pose, even if the observed leg directions did (i.e. there was not a global diffeomorphism between the observation space and the robot space).

Recently, it was shown that the visual servoing of the leg directions of the Gough-Stewart platform and the Adept Quattro with 3 translational degrees of freedom was equivalent to controlling other virtual robots that have assembly modes and singular configurations different from those of the real ones. These hidden robot models are tangible visualizations of the mapping between the observation space and the real robots Cartesian space. Thanks to this concept, all the aforementioned points were answered for the mentioned robots.

In this paper, the concept of hidden robot model is generalized for any type of parallel robots controlled using visual servos based on the observation of the leg directions. It is shown that the concept of hidden robot model is a powerful tool that gives useful insights about the visual servoing of robots using leg direction observation. With the concept of hidden robot model, the singularity problem of the controller can be addressed and the convergence issues of the controller can be explained, understood and solved.

All these results are validated in simulations and through experiments on a Quattro robot.

Keywords: Parallel robots, visual servoing, controllability, kinematics, singularity.

1. Introduction

Parallel robots are mechanical architectures whose end-effector is linked to the fixed base by means of at least two kinematic chains [1]. Compared to serial robots, such robots are stiffer and can reach higher speeds and accelerations [2]. However, their control is troublesome because of the complex mechanical structure, highly coupled joint motions and many other factors (e.g. clearances, assembly errors, etc.) which degrade stability and accuracy.

Many research papers focus on the control of parallel mechanisms [2]. Cartesian control is naturally achieved through the use of the inverse differential kinematic model which transforms Cartesian velocities into joint velocities. It is noticeable that, in a general manner, 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.

Past research works have proven that the robot end-effector pose can be effectively estimated by vision [3, 4, 5]. The most common approach consists of the direct observation of the end-effector pose [6,

*Corresponding author
Email addresses: Sebastien.Briot@irccyn.ec-nantes.fr (Sébastien Briot), Victor.Rosenzveig@irccyn.ec-nantes.fr (Victor Rosenzveig), Philippe.Martinet@irccyn.ec-nantes.fr (Philippe Martinet), erol.ozgur@sigma-clermont.fr (Erol Özgür), nicolas.bouton@sigma-clermont.fr (Nicolas Bouton)

7, 8]. However, some applications prevent the observation of the end-effector of a parallel mechanism by vision [9, 10, 11, 12]. For instance, it is not wise to imagine observing the end-effector of a machine-tool while it is generally not a problem to observe its legs that are most often designed with slim and rectilinear rods [2].

A first step in this direction was made in [13] where vision was used to derive a visual servoing scheme based on the observation of a Gough-Stewart (GS) parallel robot [14]. In that method, the leg directions were chosen as visual primitives and control was derived based on their reconstruction from the image. By stacking the observation matrices corresponding to the observation of several legs, a control scheme was derived and it was then shown that such an approach allowed the control of the observed robot. After these preliminary works, the approach was extended to the control of the robot directly in the image space by the observation of the leg edges (from which the leg direction can be extracted), which has proven to exhibit better performances in terms of accuracy than the previous approach [15]. The approach was applied to several types of robots, such as the Adept Quattro and other robots of the same family [16, 17].

The proposed control scheme was not usual in visual servoing techniques, in the sense that in the controller, both robot kinematics and observation models linking the Cartesian space to the leg direction space are involved. As a result, some surprising results were obtained:

1. the observed robot which is composed of n legs could be controlled in most cases using the observation of only m leg directions ($m < n$), knowing the fact that the minimal number of observed legs should be, for 3D unit vectors characterizing the leg directions, an integer greater than $n/2$
2. in some cases, the robot does not converge to the desired end-effector pose (even if the observed leg directions did)

without finding some concrete explanations to these points. Especially, the last point showed that it may be possible that a global diffeomorphism between the Cartesian space and the leg direction space does not exist, but no formal proof was given.

In parallel, some important questions were never answered, such as:

3. How can we be sure that the stacking of the observation matrices cannot lead to local minima (for which the error in the observation space is non zero while the robot platform cannot move [18]) in the Cartesian space?
4. Are we sure that there is no singularity in the mapping between the leg direction space and the Cartesian space?

All these points were never answered because of the *lack of existing tools* able to analyze the intrinsic properties of the controller. Additionally, we would like to point out that the understanding of the singularity cases of the mapping used in the controller is of the utmost because these singularities leads to the loss of controllability of the robot [19], and thus the define the boundaries of the reachable workspace for the controller. As a result, the accessible workspace for the robot controlled by leg observation is the intersection of two workspaces: (1) the singularity-free workspace of the robot and (2) the workspace free of singularities linked to the mapping between the leg direction space and the Cartesian space.

Recently, two of the authors of the present paper have demonstrated in [20] that these points could be explained by considering that the visual servoing of the leg directions of the GS platform was equivalent to controlling another robot "hidden" within the controller, the 3–$\underline{U}PS$[1] that has assembly modes and singular configurations different from those of the GS platform.

In both cases, considering this hidden robot model allowed the finding of a minimal representation for the leg-observation-based control of the studied robots that is linked to a virtual hidden robot which is a tangible visualization of the mapping between the observation space and the real robot Cartesian space. The hidden robot model:

1. can be used to explain why the observed robot which is composed of n legs can be controlled using the observation of only m leg directions ($m < n$),

[1] In the following of the paper, R, P, U, S, Π will stand for passive revolute, prismatic, universal, spherical and planar parallelogram joint [21], respectively. If the letter is underlined, the joint is considered active.

2. can be used to prove that there does not always exist a global diffeomorphism between the Cartesian space and the leg direction space, but can also bring solutions for avoiding to converge to a non desired pose,
3. simplifies the singularity analysis of the mapping between the leg direction space and the Cartesian space by reducing the problem to the singularity analysis of a new robot.
4. can be used to certify that the robot will not converge to local minima, through the application of tools developed for the singularity analysis of robots.

Thus, the concept of hidden robot model, associated with mathematical tools developed by the mechanical design community, is a powerful tool able to analyze the intrinsic properties of some controllers developed by the visual servoing community. Moreover, this concept shows that in some visual servoing approaches, stacking several interaction matrices to derive a control scheme without doing a deep analysis of the intrinsic properties of the controller is clearly not enough. Further investigations are required.

Therefore, in this paper, the generalization of the concept of hidden robot model is presented and a general way to find the hidden robots corresponding to any kind of robot architecture is explained. It will be shown that the concept of hidden robot model is a powerful tool that gives useful insights about the visual servoing of robots using leg direction observation. With the concept of hidden robot model, the singularity problem of the mapping between the space of the observed robot links and the Cartesian space can be addressed, and above all, it is possible to give and certify information about the controllability of the observed robots using the proposed controller. Therefore, with the hidden robot concept, we are able to understand and find the minimal representation for the control of parallel robots via leg observation, i.e. to find what are the necessary (minimal) information to use in the controller in order to allow the full control of the robot.

The paper is decomposed as follows. Section 2 makes some brief recalls on the visual servoing of parallel robots using leg observations. Then, Section 3 presents the concept of hidden robot model and generalizes the approach for any type of parallel robots. In Section 4, some examples of typical classes of parallel robots are studied: the planar parallel robots, the n-Pod family (i.e. GS platform-like robots), and the Delta-like robots. Simulations and experimental validations on the Adept Quattro are presented in Section 5. Finally, our conclusions are written in Section 6.

2. Background on visual servoing of parallel robots using leg observations

2.1. Line modeling

A line $\mathcal{L}$ in space, expressed in the camera frame, is defined by its Bi-normalized Plücker coordinates [22]:

$$\mathcal{L} \equiv \left({}^c\underline{\mathbf{u}}, {}^c\underline{\mathbf{n}}, {}^cn\right) \tag{1}$$

where ${}^c\underline{\mathbf{u}}$ is the unit vector giving the spatial orientation of the line[2], ${}^c\underline{\mathbf{n}}$ 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 ${}^cn{}^c\underline{\mathbf{n}} = {}^c\mathbf{P} \times {}^c\underline{\mathbf{u}}$ where ${}^c\mathbf{P}$ is the position of any point P on the line, expressed in the camera frame. Notice that, using this notation, the well-known (normalized) Plücker coordinates [23, 2] are the couple $\left({}^c\underline{\mathbf{u}}, {}^cn{}^c\underline{\mathbf{n}}\right)$.

The projection of such a line in the image plane, expressed in the camera frame, has for characteristic equation [22]:

$${}^c\underline{\mathbf{n}}^T{}^c\mathbf{p} = 0 \tag{2}$$

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

[2] In the following of the paper, the superscript before the vector denotes the frame in which the vector is expressed ("b" for the base frame, "c" for the camera frame and "p" for the pixel frame). If there is no superscript, the vector can be written in any frame.

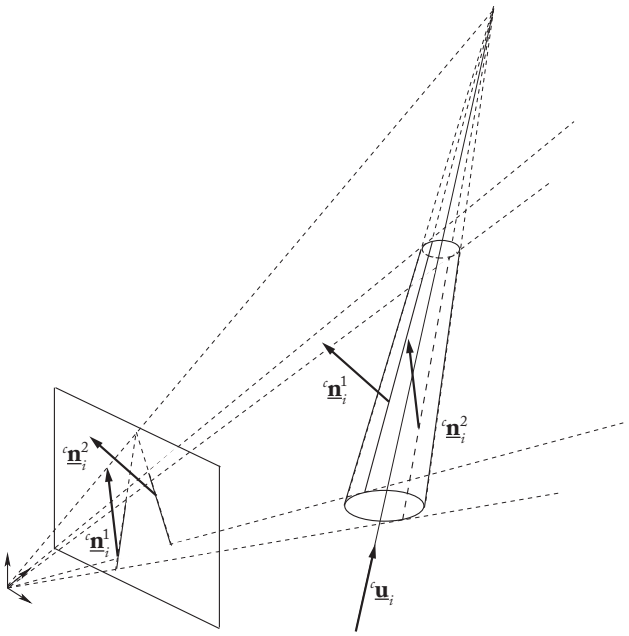


Figure 1: Projection of a cylinder in the image

2.2. Cylindrical leg observation

The legs of parallel robots have usually cylindrical cross-sections [2]. The edges of the i-th cylindrical leg are given, in the camera frame, by [15] (Fig 1):

$$^c\underline{\mathbf{n}}_i^1 = -\cos\theta_i {}^c\underline{\mathbf{h}}_i - \sin\theta_i {}^c\underline{\mathbf{u}}_i \times {}^c\underline{\mathbf{h}}_i \tag{3}$$

$$^c\underline{\mathbf{n}}_i^2 = +\cos\theta_i {}^c\underline{\mathbf{h}}_i - \sin\theta_i {}^c\underline{\mathbf{u}}_i \times {}^c\underline{\mathbf{h}}_i \tag{4}$$

where $\cos\theta_i = \sqrt{{}^ch_i^2 - R_i^2}\big/{}^ch_i$, $\sin\theta_i = R_i/{}^ch_i$ and $\left({}^c\underline{\mathbf{u}}_i, {}^c\underline{\mathbf{h}}_i, {}^ch_i\right)$ are the Bi-normalized Plücker coordinates of the cylinder axis and R_i is the cylinder radius.

It was also shown in [15] that the leg orientation, expressed in the camera frame, is given by

$$^c\underline{\mathbf{u}}_i = \frac{^c\underline{\mathbf{n}}_i^1 \times {}^c\underline{\mathbf{n}}_i^2}{\left\|{}^c\underline{\mathbf{n}}_i^1 \times {}^c\underline{\mathbf{n}}_i^2\right\|} \tag{5}$$

Let us remark now that each cylinder edge is a line in space, with Bi-normalized Plücker expressed in the camera frame $({}^c\underline{\mathbf{u}}_i, {}^c\underline{\mathbf{n}}_i^j, {}^cn_i^j)$ (Fig 1).

2.3. Leg direction based visual servoing

The proposed control approach was to servo the leg directions ${}^c\underline{\mathbf{u}}_i$ [13]. Some brief recalls on this type of controller are done below.

2.3.1. Interaction matrix

Visual servoing is based on the so-called interaction matrix $\mathbf{L}^T$ [24] which relates the instantaneous relative motion $T_c = {}^c\boldsymbol{\tau}_c - {}^c\boldsymbol{\tau}_s$ between the camera and the scene, to the time derivative of the vector s of all the visual primitives that are used through:

$$\dot{s} = \mathbf{L}_{(s)}^T T_c \tag{6}$$

where ${}^c\boldsymbol{\tau}_c$ and ${}^c\boldsymbol{\tau}_s$ are respectively the kinematic screw of the camera and the scene, both expressed in $\mathcal{R}_c$, i.e. the camera frame.

In the case where we want to directly control the leg directions ${}^c\underline{\mathbf{u}}_i$, and if the camera is fixed, (6) becomes:

$$^c\dot{\underline{\mathbf{u}}}_i = \mathbf{M}_i^{T\,c}\boldsymbol{\tau}_c \tag{7}$$

where $\mathbf{M}_i^T$ is the interaction matrix for the leg i.

2.3.2. Control

For the visual servoing of a robot, one achieves exponential decay of an error $e(s, s_d)$ between the current primitive vector s and the desired one s_d using a proportional linearizing and decoupling control scheme of the form:

$$T_c = \lambda \hat{\mathbf{L}}_{(s)}^{T+} e(s, s_d) \tag{8}$$

where T_c is used as a pseudo-control variable and the superscript "+" corresponds to the matrix pseudo-inverse.

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:

$$\mathbf{e}_i = {}^c\underline{\mathbf{u}}_i \times {}^c\underline{\mathbf{u}}_{di} \tag{9}$$

where ${}^c\underline{\mathbf{u}}_{di}$ is the desired value of ${}^c\underline{\mathbf{u}}_i$.

It can be proven that, for spatial parallel robots, matrices $\mathbf{M}_i$ are in general of rank 2 [13] (for planar parallel robots, they are of rank 1). As a result, for spatial robots with more than 2 *dof*, the observation of several independent legs is necessary to control the end-effector pose. An interaction matrix $\mathbf{M}^T$ can then obtained by stacking k matrices $\mathbf{M}_i^T$ of k legs.

Finally, a control is chosen such that $\mathbf{e}$, the vector stacking the errors $\mathbf{e}_i$ associated to of k legs ($k = 3...6$), decreases exponentially, i.e. such that

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \tag{10}$$

Then, introducing $\mathbf{L}_i^T = -\left[{}^c\underline{\mathbf{u}}_{di}\right]_\times \mathbf{M}_i^T$, where $\left[{}^c\underline{\mathbf{u}}_{di}\right]_\times$ is the cross product matrix associated with the vector ${}^c\underline{\mathbf{u}}_{di}$, the combination of (9), (7) and (10) gives

$${}^c\boldsymbol{\tau}_c = -\lambda \mathbf{L}^{T+} \mathbf{e} \tag{11}$$

where $\mathbf{L}^T$ can be obtained by stacking the matrices $\mathbf{L}_i^T$ of k legs. The conditions for the rank deficiency of matrix $\mathbf{L}^T$, as well as the conditions that lead to local minima [18] of the Eq. (11) are discussed in Section 3.

This expression can be transformed into the control joint velocities:

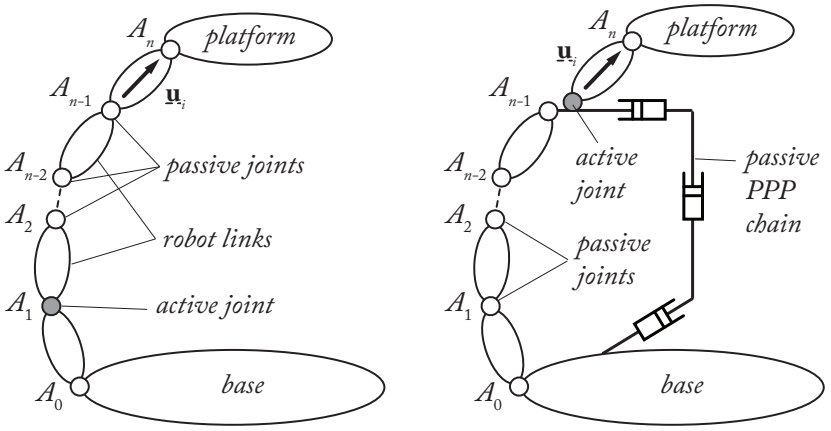
$$\dot{\mathbf{q}} = -\lambda {}^c\mathbf{J}^{inv} \mathbf{L}^{T+} \mathbf{e} \tag{12}$$

where ${}^c\mathbf{J}^{inv}$ is the inverse Jacobian matrix of the robot relating the end-effector twist to the actuator velocities, i.e. ${}^c\mathbf{J}^{invc}\boldsymbol{\tau}_c = \dot{\mathbf{q}}$.

In the next Section, it is shown that the equations used in the controller, characterizing the mapping between the observation (leg-direction) space and the Cartesian space, can indeed be related to a virtual architecture hidden within the controller. Understanding and using the models of these hidden robots is mandatory for being able to analyze the controllability of parallel robots using the proposed visual servoing approach.

3. The concept of hidden robot model

The concept of hidden robot model has been first introduced in [20] for the visual servoing of the GS platform. In this paper, it has been demonstrated that the leg direction based visual servoing of such robots intrinsically involves the appearance of a hidden robot model, which has assembly modes and singularities different from the real robot. It was shown that the concept of hidden robot model fully explains the possible non convergence of the observed robot to the desired final pose and that it considerably simplifies the singularity analysis of the mapping involved in the controller. The aim of this Section is to generalize and extend the concept to any class of robots.



(a) A general robot leg

(b) its corresponding hidden robot leg when the vector $\underline{\mathbf{u}}_i$ is observed

Figure 2: A general robot leg and its corresponding hidden robot leg when the vector $\underline{\mathbf{u}}_i$ is observed

3.1. Statement of the problem

The concept of hidden robot model comes from the following observation: in the classical control approach, the encoders measure the motion of the actuator; in the previously described control approach (Section 2), the leg directions or leg edges are observed. So, in a reciprocal manner, one could wonder to what kind of virtual actuators such observations correspond, and as a result, what could be the virtual architecture hidden inside the controller of a given robot. The main objective of this Section is to give a general answer to these questions.

However, we want to claim that our objective is not to prove the unicity of the virtual robot hidden in the controller. Indeed, several robot architectures could be found for the same controller, but all of them will be equivalent in terms of geometry and kinematics (same input-output relationships, same singularities). So, it is not necessary to find an architecture solution of an optimal problem, but it is sufficient to find one architecture having the desired geometric and kinematic properties in order to solve the singularity problem of the controller. This is what is shown below with a rather simple and intuitive approach.

3.2. How to define the legs of the hidden robots

Let us consider a general leg for a parallel robot in which the direction $\underline{\mathbf{u}}_i$ of a segment is observed (Fig. 2(a) – in this figure, the last segment is considered observed, but the following explanations can be generalized to any segment located in the leg chain). In what follows, we only consider that we observe the leg direction $\underline{\mathbf{u}}_i$, and not the leg edges in the image space, as the leg edges are only used as a measure of $\underline{\mathbf{u}}_i$. So *the problem is the same*, except in the fact that we must consider the singularity of the mapping between the edges and $\underline{\mathbf{u}}_i$, but this problem is well handled: these singularities appear when $\underline{\mathbf{n}}_i^1$ and $\underline{\mathbf{n}}_i^2$ are collinear, i.e. the cylinders are at infinity [15].

In the general case, the unit vector $\underline{\mathbf{u}}_i$ can obviously be parameterized by two independent coordinates, that can be two angles, for example the angles α and β of Fig. 3 defined such that $\cos\alpha = \underline{\mathbf{x}} \cdot \underline{\mathbf{v}} = \underline{\mathbf{y}} \cdot \underline{\mathbf{w}}$ (where $\underline{\mathbf{v}}$ and $\underline{\mathbf{w}}$ are defined such that $\underline{\mathbf{z}} \cdot \underline{\mathbf{v}} = \underline{\mathbf{z}} \cdot \underline{\mathbf{w}} = 0$) and $\cos\beta = \underline{\mathbf{u}} \cdot \underline{\mathbf{x}}$. Thus α is the angle of the first rotation of the link direction $\underline{\mathbf{u}}_i$ around $\underline{\mathbf{z}}$ and β is the angle of the second rotation around $\underline{\mathbf{v}}$.

It is well known that a U joint is able to orientate a link around two orthogonal axes of rotation, such as $\underline{\mathbf{z}}$ and $\underline{\mathbf{v}}$. Thus U joints can be the virtual actuators with generalized coordinates α and β we are looking for. Of course, other solutions can exist, but U joints are the simplest ones.

If a U joint is the virtual actuator that makes the vector $\underline{\mathbf{u}}_i$ move, it is obvious that:

- if the value of $\underline{\mathbf{u}}_i$ is fixed, the U joint coordinates α and β must be constant, i.e. *the actuator must be blocked*,
- if the value of $\underline{\mathbf{u}}_i$ is changing, the U joint coordinates α and β must also vary.

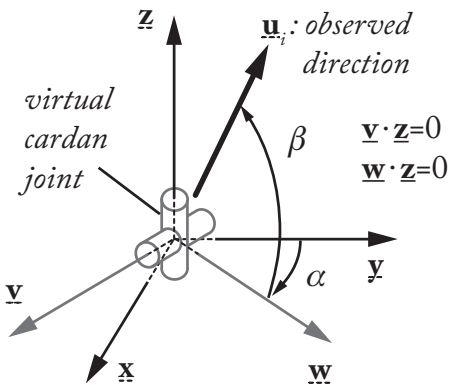


Figure 3: Parameterization of a unit vector $\underline{\mathbf{u}}_i$ with respect to a given frame $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ and $\underline{\mathbf{z}}$

As a result, to ensure the aforementioned properties for α and β if $\underline{\mathbf{u}}_i$ is expressed in the base or camera frame (but the problem is identical as the camera is considered fixed on the ground), vectors $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$ and $\underline{\mathbf{z}}$ of Fig. 3 must be the vectors defining the base or camera frame. Thus, in terms of properties for the virtual actuator, this implies that the first $\underline{U}$ joint axis must be constant w.r.t. the base frame, i.e. the $\underline{U}$ joint must be attached to a link *performing a translation w.r.t. the base frame*[3].

However, in most of the cases, the real leg architecture is not composed of U joints attached on links performing a translation w.r.t. the base frame. Thus, the architecture of the hidden robot leg must be modified w.r.t. the real leg such as depicted in Fig. 2(b). The $\underline{U}$ joint must be mounted on a passive kinematic chain composed of at most 3 orthogonal passive P joints that ensures that the link on which is it attached performs a translation w.r.t. the base frame. This passive chain is also linked to the segments before the observed links so that they do not change their kinematic properties in terms of motion. Note that:

- it is necessary to fix the PPP chain on the preceding leg links because the information given by the vectors $\underline{\mathbf{u}}_i$ is not enough for rebuilding the full platform position and orientation: it is also necessary to get information on the location of the anchor point A_{n-1} of the observed segment [15]. This information is kept through the use of the PPP chain fixed on the first segments;

- 3 P joints are only necessary if and only if the point A_{n-1} describes a motion in the 3D space; if not, the number of P joints can be decreased: for example, in the case of the GS platform presented in [20], the U joint of the leg to control was located on the base, i.e. there was no need to add passive P joints to keep the orientation of its first axis constant;

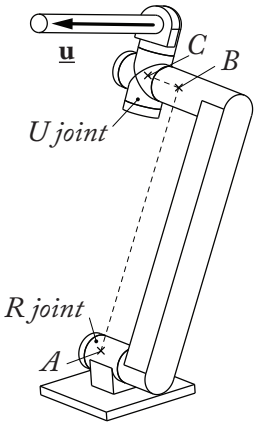
- when the vector $\underline{\mathbf{u}}_i$ is constrained to move in a plane such as for planar legs, the virtual actuator becomes an R joint which must be mounted on the passive PPP chain (for the same reasons as mentioned previously).

For example, let us have a look at the $\underline{R}U$ leg with one actuated $\underline{R}$ joint followed by a U joint of Fig. 4(a). Using the previous approach, its virtual equivalent leg should be an $\{R\text{–}PP\}\text{–}\underline{U}$ leg (Fig. 4(b)), i.e. the $\underline{U}$ joint able to orientate the vector $\underline{\mathbf{u}}_i$ is mounted on the top of a $R\text{–}PP$ chain that can guarantee that:

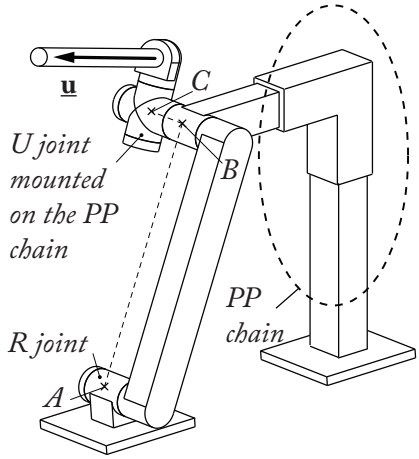
1. the link on which the $\underline{U}$ joint is attached performs a translation w.r.t. the base frame,
2. the point C (i.e. the centre of the U joint) evolves on a circle of radius l_{AB}, like the real leg.

It should be noticed that, in several cases for robots with a lower mobility (i.e. spatial robots with a number of *dof* less than 6, or planar robots with a number of *dof* less than 3), the last joint that links the leg to the platform should be changed so that, if the number of observed legs is inferior to the number of real legs, the hidden robot keeps the same number of controlled *dof* (see Section 4.2.2).

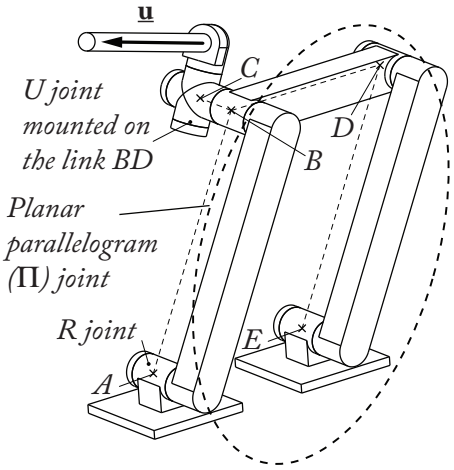
[3] In the case where the camera is not mounted on the frame but on a moving link, the virtual $\underline{U}$ joint must be attached on a link performing a translation w.r.t. the considered moving link.



(a) A $\underline{R}U$ leg



(b) Virtual $\{R\text{–}PP\}\text{–}\underline{U}$ leg



(c) Virtual $\Pi\underline{U}$ leg

Figure 4: A $\underline{R}U$ leg and two equivalent solutions for its hidden leg

It should also be mentioned that we have presented above the most general methodology that is possible to propose, but it is not the most elegant way to proceed. In many cases, a hidden robot leg architecture can be obtained such that less modifications w.r.t the real leg are achieved. For example, the *R–PP* chain of the hidden robot leg $\{R\text{–}PP\}\text{–}\underline{U}$ (Fig. 4(b)) could be equivalently replaced by a planar parallelogram (Π) joint without changing the aforementioned properties of the $\underline{U}$ virtual actuator (Fig. 4(c)), i.e. only one additional joint is added for obtaining the hidden robot leg (note that we consider that a Π joint, even if composed of several pairs, can be seen as one single joint, as in [21]).

Anyway, as mentioned above, the objective of this part is not to find an architecture solution of an optimal problem (optimal with respect to design complexity [21], for instance) but it was to find one architecture having the desired geometric and kinematic properties in order to solve the singularity problem of the controller. This is what was done below with a rather simple and intuitive approach.

It should be noted that the same results could be also demonstrated with more rigorous Type synthesis approaches [25, 26, 27]. However, adding such methodologies in the present paper would have not provided any further explanations of the problem and would have make it longer and more unclear.

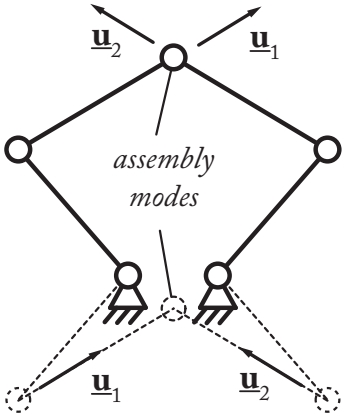


Figure 5: Two configurations of a five bar mechanism for which the directions $\underline{\mathbf{u}}_i$ are identical (for $i = 1, 2$)

3.3. How to use the hidden robot models for analyzing the controllability of the servoed robots

The aim of this Section is to show how to use the hidden robots for answering points 1 to 4 enumerated in the introduction of the paper.

Point 1: the hidden robot model can be used to explain why the observed robot which is composed of n legs could be controlled in most cases using the observation of only m leg directions ($m < n$), knowing the fact that the minimal number of observed legs should be, for 3D unit vectors, an integer greater than $n/2$.

To answer this point, let us consider a general parallel robot composed of 6 legs (one actuator per leg) and having six *dof*. Using the approach proposed in Section 3.2, each observed leg will lead to a modified virtual leg with at least one actuated $\underline{U}$ joint that has two degrees of actuation. For controlling 6 *dof*, only 6 degrees of actuation are necessary, i.e. three actuated $\underline{U}$ are enough (as long as the motions of the $\underline{U}$ joints are not correlated, i.e. the robot is fully actuated). Thus, in a general case, only three legs have to be observed to fully control the platform *dof*.

Point 2: the hidden robot model can be used to prove that there does not always exist a global diffeomorphism between the Cartesian space and the leg direction space, but can also bring solutions for avoiding to converge to a non desired pose.

Here, the answer comes directly from the fact that the real controlled robot may have a hidden robot model with different geometric and kinematics properties. This means that the hidden robot may have assembly modes and singular configurations different from those of the real robot. If the initial and final robot configurations are not included in the same aspect (i.e. a workspace area that is singularity-free and bounded by singularities [2]), the robot won't be able to converge to the desired pose, but to a pose that corresponds to another assembly mode that has the same leg directions as the desired final pose (see Fig. 5).

Solutions for avoiding to converge to a non-desired pose are given in Appendix Appendix A.

Point 3: the hidden robot model simplifies the singularity analysis of the mapping between the leg direction space and the Cartesian space by reducing the problem to the singularity analysis of a new robot.

The interaction matrix $\mathbf{M}^T$ involved in the controller gives the value of ${}^c\underline{\dot{\mathbf{u}}}$ as a function of ${}^c\boldsymbol{\tau}_c$. Thus, $\mathbf{M}^T$ is the inverse Jacobian matrix of the hidden robot (and, consequently, $\mathbf{M}^{T+}$ is the hidden robot Jacobian matrix). Except in the case of decoupled robots [28, 29, 30], the Jacobian matrices of parallel robots are not free of singularities.

Considering the input/output relations of a robot, three different kinds of singularity can be observed [31][4]:

- the *Type 1 singularities* that appear when the robot Jacobian matrix is rank-deficient; in such configurations, any motion of the actuator that belongs to the kernel of the Jacobian matrix is not able to produce a motion of the platform,
- the *Type 2 singularities* that occur when the robot inverse Jacobian matrix is rank-deficient; in such configurations, any motion of the platform that belongs to the kernel of the inverse Jacobian matrix is not able to produce a motion of the actuator. And, reciprocally, near these configurations, a small motion of the actuators lead to large platform displacements, i.e. the accuracy of the robot becomes very poor,
- the *Type 3 singularities* that appear when both the robot Jacobian and inverse Jacobian matrices are rank-deficient.

Thus,

- finding the condition for the rank-deficiency of $\mathbf{M}^T$ is equivalent to find the Type 2 singularities of the hidden robot,
- finding the condition for the rank-deficiency of $\mathbf{M}^{T+}$ is equivalent to find the Type 1 singularities of the hidden robot.

Since a couple of decades ago, many tools have been developed by the mechanical design community for finding the singular configurations of robots. The interested reader could refer to [33, 2, 34, 35] and many other works on the Grassmann Geometry and Grassmann-Cayley Algebra for studying the singular configurations problem. In what follows in the paper, these tools are used but only the final results concerning the singular configuration conditions are given.

Point 4: the hidden robot model can be used to certify that the robot will not converge to local minima.

The robot could converge to local minima if the matrix $\mathbf{L}^{T+}$ of (11) is rank deficient. A necessary and sufficient condition for the rank deficiency of this matrix is that the $\mathbf{M}^{T+}$ is rank deficient, i.e. the hidden robot model encounters a Type 1 singularity. As mentioned above, many tools have been developed by the mechanical design community for finding the singular configurations of robots and solutions can be provided to ensure that the hidden robot model does not meet any Type 1 singularity (see also the Appendix Appendix A).

4. Hidden robot models of some known parallel robot families

Let us now present the hidden robot models of some well known families of parallel robots and deal with their forward kinematics and singular configurations.

It should be noted that, in the singularity analysis of the considered robots, only the Type 2 singular configurations are detailed as Type 1 singularity always appear when at least one leg is fully stretched or folded (workspace boundary). Therefore, these conditions are not recalled.

4.1. Application to planar parallel robots

4.1.1. The hidden robot legs of planar parallel robots

The usual planar parallel manipulators (*ppm*) are composed of planar serial chains with at most three 1-*dof* joints, respectively, among which one is actuated. As mentioned in [2], using the different possible combinations of *R* and *P* joints, only 10 different serial chains, that lead to robots that can be actuated, can

[4]There exist other types of singularities, such as the constraint singularities [32], but they are due to passive constraint degeneracy only, and are not involved in the mapping between the leg directions space and the robot controlled Cartesian coordinate space.

be obtained. These chains are represented in Table 1 (in this table and the following pictures, the gray pairs denote the actuated joints).

Now, using the approach presented in Section 3.2, and considering that the direction $\underline{\mathbf{u}}_i$ of the last segment of each leg is observed, one can find the hidden robot leg corresponding to this observation (Table 1).

Table 1: The 10 possible architectures for the legs of *ppm* and their equivalent hidden robot leg for the visual servoing using leg directions

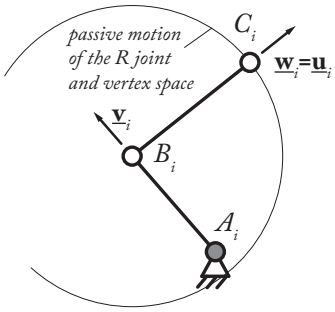
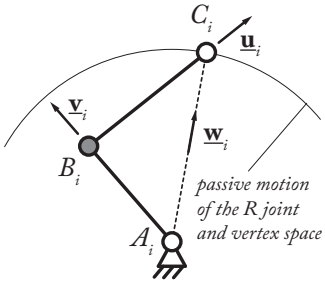
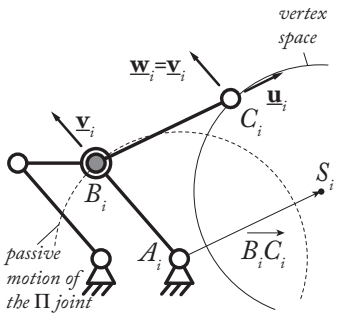
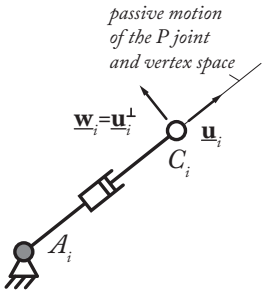
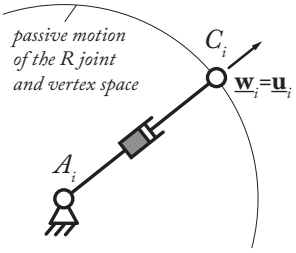
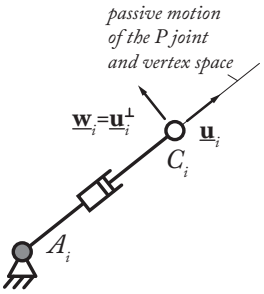
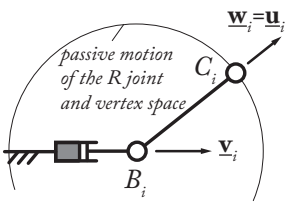
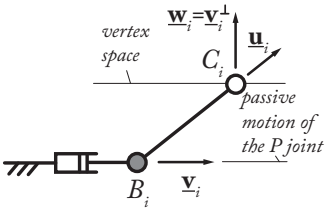
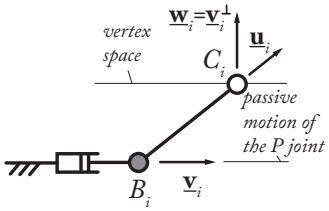
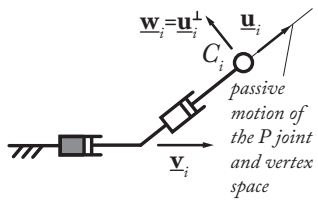
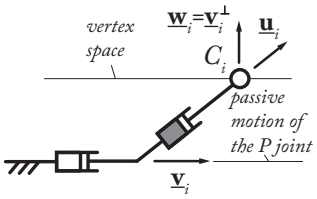
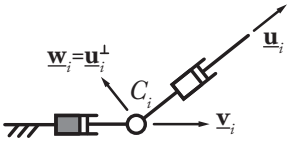
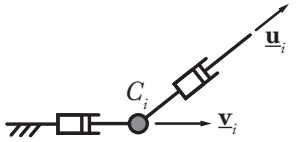
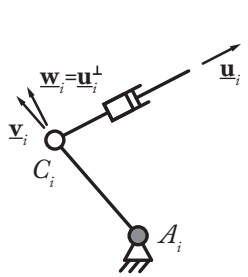
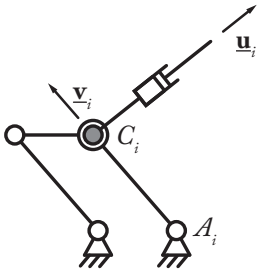
Real leg architecture		Hidden robot leg architecture
for $\underline{R}RR$	and $R\underline{R}R$ legs $\Longrightarrow$	a $\Pi\underline{R}R$ leg
		
for $\underline{R}PR$	and $R\underline{P}R$ legs $\Longrightarrow$	a $\underline{R}PR$ leg
		
for $\underline{P}RR$	and $P\underline{R}R$ legs $\Longrightarrow$	a $P\underline{R}R$ leg
		
for $\underline{P}PR$	and $P\underline{P}R$ legs $\Longrightarrow$	*no equivalent hidden robot leg*
		
for $\underline{P}RP$ legs $\Longrightarrow$		a $P\underline{R}P$ leg
		

Table 1: The 10 possible architectures for the legs of *ppm* and their equivalent hidden robot leg for the visual servoing using leg directions

Real leg architecture	Hidden robot leg architecture
for $\underline{R}RP$ legs $\Longrightarrow$ 	a $\Pi\underline{R}P$ leg 

From Table 1, the following information can be extracted:

- for $\underline{R}PR$ and $P\underline{R}R$ legs, the hidden robot legs are the same as the real ones;
- $\underline{P}RP$ legs lead to $P\underline{R}P$ hidden robot legs; and robots made of $P\underline{R}P$ legs are well known not to be controllable [2]. A similar result appears for $\underline{R}RP$ legs that lead to $\Pi\underline{R}P$ hidden robot legs.
- the last element of $\underline{P}PR$ and $PP\underline{R}$ having a constant direction $\underline{\mathbf{u}}_i$, robots made of such legs cannot be controlled using leg direction observation. As a result, they don't have an equivalent hidden robot model.

Thus, using the concept of hidden robot leg and hidden robot model, the problem of the robot controllability can be directly addressed without any mathematical derivations.

The next Section presents the hidden robot models of the 2 and 3-*dof* controllable robot with a symmetric leg arrangement and made of the legs presented in Table 1.

4.1.2. The hidden robot models of planar parallel robots

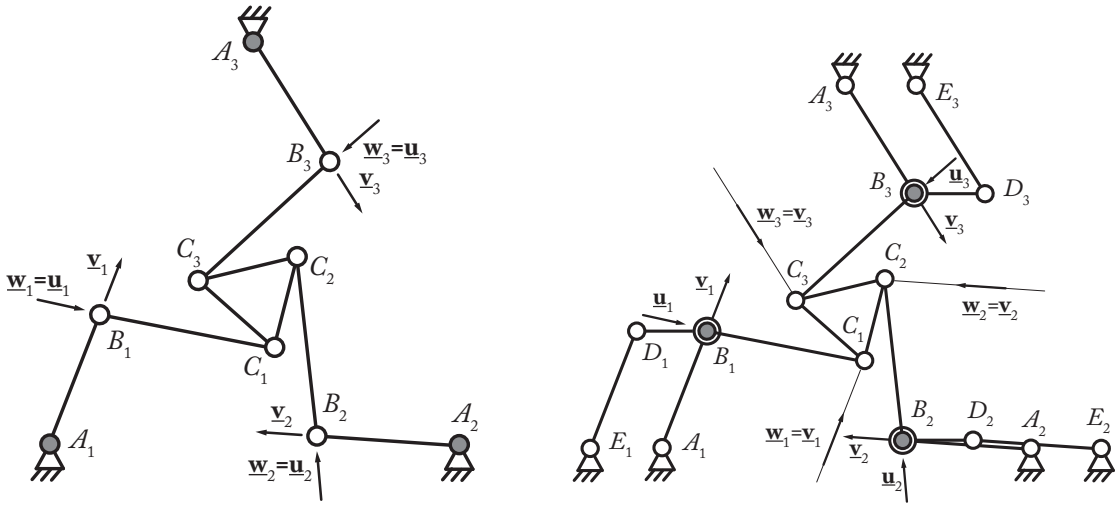
Using the results of the previous Section, 12 *ppm* with symmetric leg arrangement (i.e. with identical leg architectures) and that can be controlled using the leg direction observation can be found:

- for manipulators with 2 *dof*: $\underline{R}RRR\underline{R}$, $R\underline{R}R\underline{R}R$, $\underline{R}PRP\underline{R}$, $R\underline{P}R\underline{P}R$, $\underline{P}RRR\underline{P}$, $P\underline{R}R\underline{R}P$ robots;
- for manipulators with 3 *dof*: 3–$\underline{R}RR$, 3–$R\underline{R}R$, 3–$\underline{R}PR$, 3–$R\underline{P}R$, 3–$\underline{P}RR$, 3–$P\underline{R}R$ robots.

Their architectures are well-known and, as they can also be easily deduced from the leg arrangement given in Tab. 1, their schematics are not depicted again.

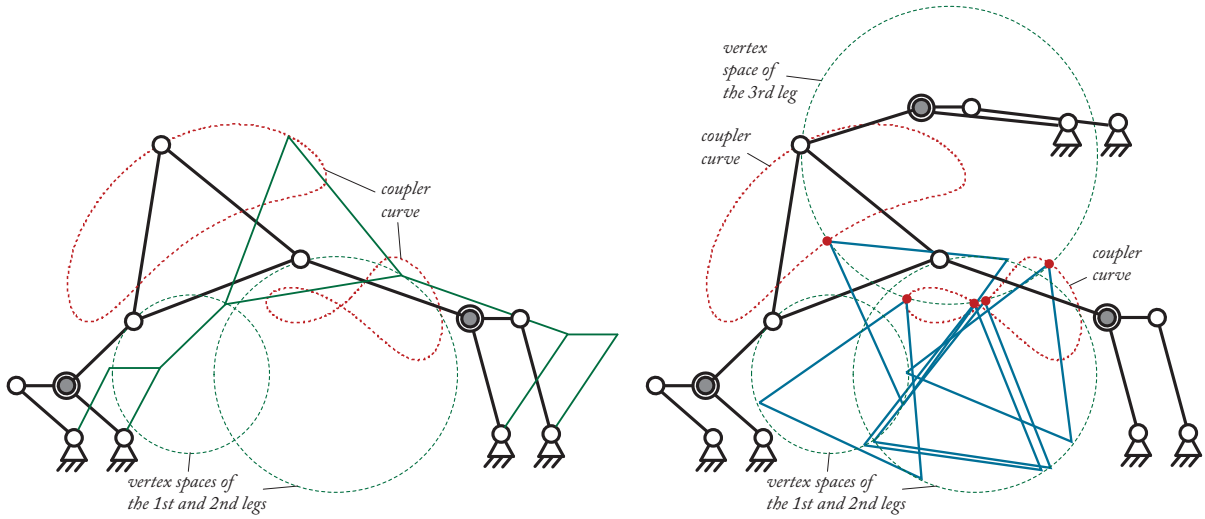
However, for illustrating this Section, let us present the forward kinematic problem (*fkp*) and singularity analysis of the hidden robot model of the 3–$\underline{R}RR$ robot, when controlled using leg direction observation (Fig. 6(a)). Using the results of Table 1, it can be found that its equivalent hidden robot model is a 3–$\Pi\underline{R}R$ robot (Fig. 6(b)). Each of its legs is composed of a passive planar parallelogram (Π joint) which is able to maintain constant the orientation of the links B_iD_i w.r.t. the base and of an $\underline{R}R$ chain which is mounted on the link B_iD_i.

Forward kinematics and assembly modes. Using the usual methodology [2], all the solutions to the *fkp* are at the intersections of the coupler curve (which represents the displacement loci of one platform extremity when one of the leg is disassembled, the actuators of the two other being fixed (see Fig. 7(a))) with the vertex space of the disassembled leg (that represents the passive displacement of the leg tip when the actuator is fixed (see Table 1)). For the studied 3–$\Pi\underline{R}R$ robot, as the leg vertex spaces are circles (Table 1), the coupler curve is a sextic curve [2], i.e. an algebraic curve of degree 6 (in the case where the vertex spaces had been lines,



(a) 3–$\underline{R}RR$ robot kinematics (b) 3–$\Pi\underline{R}R$ robot kinematics

Figure 6: The 3–$\underline{R}RR$ robot and its hidden robot model



(a) Coupler curve when leg 3 is disconnected (b) The 6 possible assembly modes

Figure 7: Solutions of the *fkp* for a 3–$\Pi\underline{R}R$ robot

the coupler curve would have been an ellipse [36]). Thus, the solutions of the *fkp* are at the intersection points between the aforementioned circle and sextic curve. And it is shown in [2] that if a circle intersects a sextic curve, there are at most 6 intersection points. An example of possible assembly modes for the 3–$\Pi\underline{R}R$ robot are presented in Fig. 7(b).

It should be mentioned that, even if for the 3–$\underline{R}RR$ (and as a consequence, for the 3–$R\underline{R}R$), the hidden robot has 6 assembly modes, for the other 10 robots cited at the beginning of Section 4.1.2, the maximal number of assembly modes is 2 (see Appendix Appendix A), i.e. the control approach based on the observation of the leg direction allows most of the time the decrease of complexity for the *fkp*.

Singular configurations. The Type 2 singular configurations of *ppm* have been deeply studied in the past and are well-known. For 3 *dof ppm* (moving in the xOy plane), the singularities appear when $\underline{\mathbf{s}}_1 \cdot \left(\underline{\mathbf{s}}_2 \times \underline{\mathbf{s}}_3\right) = 0$, where $\underline{\mathbf{s}}_i^T = \left[w_j^x \; w_j^y \; m_j^z\right]$ in which w_j^x and w_j^y are the x and y components of $\underline{\mathbf{w}}_j$ ($\underline{\mathbf{w}}_j$ corresponds to the direction of the effort applied by the actuated leg on the platform [33] – see Table 1 and Fig. 6) and m_j^z is the moment

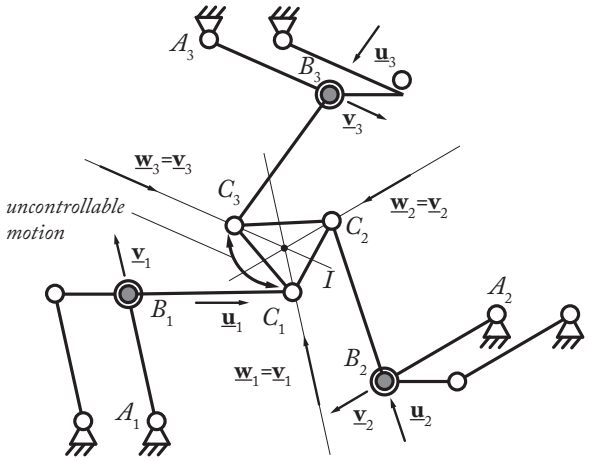


Figure 8: Example of Type 2 singularities for the 3–Π$\underline{RR}$ robot

of $\left[w_j^x\ w_j^y\right]^T$ (note that $\underline{\mathbf{w}}_j$ is always applied at point C_j) [33]. Such a condition means that the lines of Plücker coordinates $\underline{\mathbf{s}}_i$ intersect in one single point I that corresponds to the instantaneous centre of rotation of the platform in the uncontrolled *dof* (Fig. 8). This point can be at infinity: in this case, all vectors $\underline{\mathbf{w}}_j$ are parallel and the robot gets one uncontrolled translational motion in the direction orthogonal to $\underline{\mathbf{w}}_j$.

Obtained results show that singular configurations of the 3–Π$\underline{RR}$ robot are different from those singular configurations of the real 3–$\underline{R}RR$ robot, for which they appear when all lines passing through C_i of direction $\underline{\mathbf{u}}_i$ intersects in one point [33].

Note that, near the hidden robot singular configurations, the real robot accuracy will be lower. Insights about this item are given in Appendix Appendix A.

Let us now apply the concept of hidden robot models to some particular classes of spatial parallel robots.

4.2. Application to spatial parallel robots

For spatial parallel robots, due to the existence of hundreds of possible different architectures and due to the difficulty of classifying the robots by families, it is not possible to present all the hidden robot models. Thus, it is decided in this Section to show the hidden robot models of two of the best known families of parallel robots:

- the *n*-Pod family (i.e. robots such as Hexapods (or GS platforms) [14], the Tsai mechanism [37], etc.)
- the Delta-like robot family (i.e. robots such as the Delta [38], the Quattro [39], the Orthoglide [40], etc.)

For other types of robot architectures, the interested reader is referred to Section 3.2 for finding a possible hidden robot model and to [2, 34, 35] and many other works on the Grassmann Geometry and Grassmann-Cayley Algebra for studying the singular configuration problem.

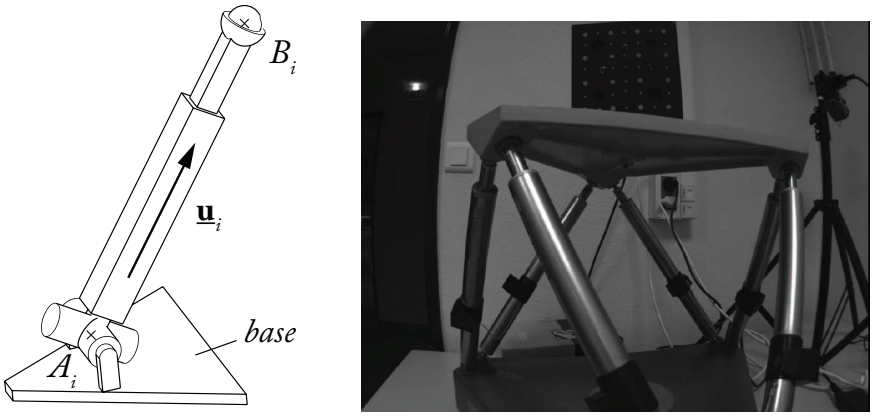
4.2.1. The n-Pod robot family

The *n*-Pod robot family regroups the robots made of *n* $U\underline{P}S$ legs (Fig. 9(a)), or some of their variations such as $U\underline{P}U$ [37] or even $R\underline{P}S$ legs [41], i.e. the legs are composed of one passive *U* or *R* joint located on the ground, followed by an active $\underline{P}$ joint and then by one passive *S* or *U* joint.

Probably the most known robots of this family are the GS platform [14] (Fig. 9(b)), the 3–$U\underline{P}U$ robots (e.g. see [37, 42]) and the 3–$R\underline{P}S$ robot [41].

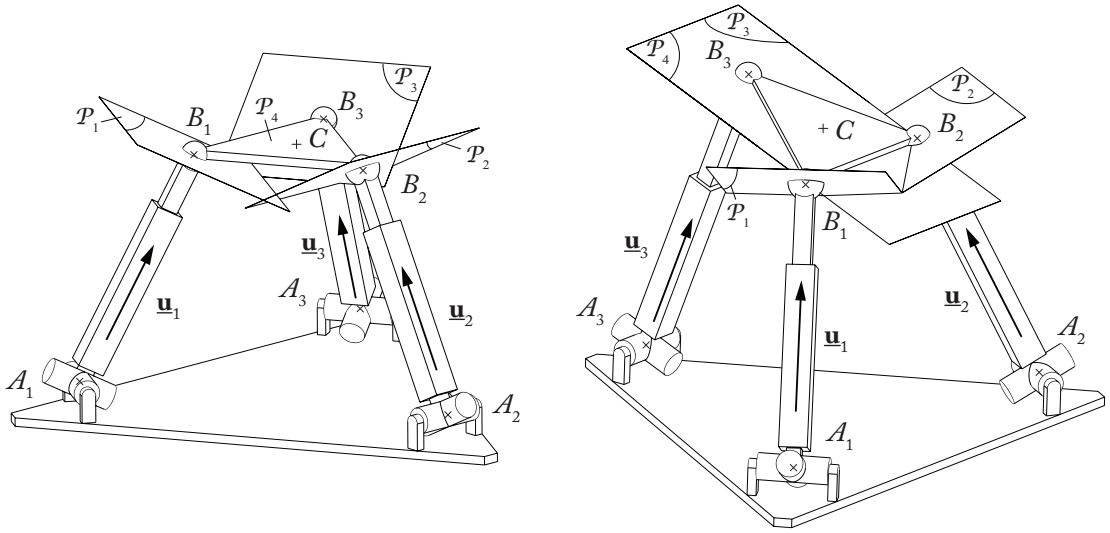
For such types of legs, the prismatic joint direction can be observed [15, 20]. From Section 3.2 and also from [20], it can be shown that the virtual equivalent legs are:

- for $U\underline{P}S$ legs: a $\underline{U}PS$ leg;
- for $U\underline{P}U$ legs: a $\underline{U}PU$ leg;



(a) *UPS* leg (b) A GS platform from DeltaLab

Figure 9: Typical legs and robots of the *n*-Pod family



(a) The 3–*UPS* robot

(b) Example of a Type 2 singularity for a 3–*UPS* robot: the platform gets an uncontrollable rotation around B_1B_2

Figure 10: Hidden robot model of the GS platform when 3 legs are observed

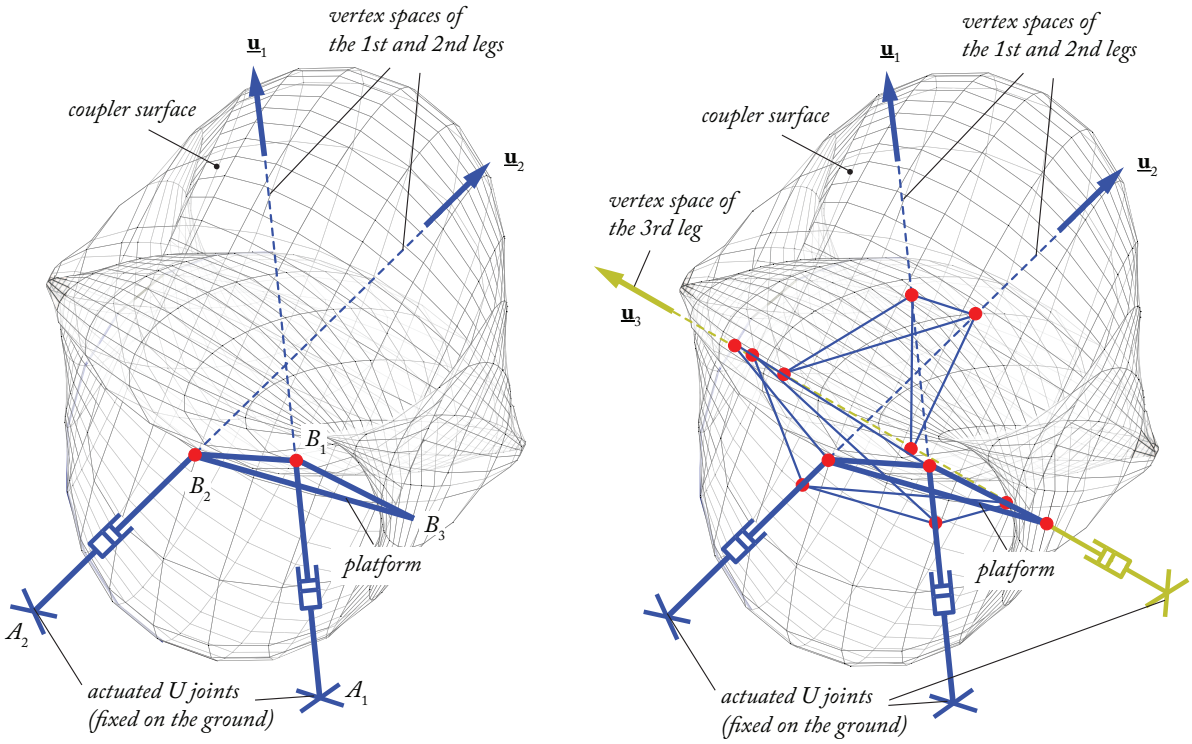
- for *RPS* legs: a *RPS* leg;

i.e. the joint fixed on the ground becomes actuated, while the prismatic joint becomes passive. Then, using the usual methodology, the *fkp* and singularity analysis can be carried out.

For illustrating this Section, let us present the *fkp* and singularity analysis for the hidden robot model of the GS platform, when controlled using leg direction observation. This analysis was already made in [20] but the present section provides additional results.

The GS platform is made of six *UPS* legs, thus its equivalent hidden robot will be made of *UPS* legs. *UPS* legs have 2 degree of actuation (the *U* joint is fully actuated), and this is the reason with only three legs have to be observed for fully controlling the GS platform using leg direction observation [20]: in this case, the hidden robot is a 3–*UPS* robot which is well known to be fully actuated (Fig. 10(a)).

Forward kinematics and assembly modes. Without loss of generality, let us consider that we analyze the 3–*UPS* robot depicted at Fig. 10(a). If leg 3 is disassembled at point B_3, as there are only four actuators for controlling the six robot mobilities, the platform gains two *dof*. The gained motion is called a *spatial Cardanic motion* [43]. This motion is defined by the fact that points B_1 and B_2 are constrained to move on the lines



(a) Coupler surface when leg 3 is disconnected

(b) Possible assembly modes (for this examples, only 6 modes can be found)

Figure 11: Solutions of the *fkp* for a 3–$\underline{U}PS$ robot

whose directions are given by $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{u}}_2$, respectively (these lines represent the vertex spaces of point B_i of the legs when the U joints are fixed and the P joints are passive), and the platform is free to rotate around the line B_1B_2. As demonstrated in [43], the surface described by point B_3 is an octic surface (Fig 11(a)), i.e. an algebraic surface of degree eight.

As B_3 also belongs to leg 3, this point is constrained to move on a line defined by the direction $\underline{\mathbf{u}}_3$ of the passive prismatic joint. As shown in [43], a line and an octic surface can have up to eight real intersection points. As a result, the 3–$\underline{U}PS$ robot can have up to eight assembly modes. An example of possible assembly modes of the 3–$\underline{U}PS$ robot is depicted in Fig. 11. Let us recall here that, in the general case, the GS platform can have up to 40 assembly modes [2] that are different from those of the 3–$\underline{U}PS$ robot.

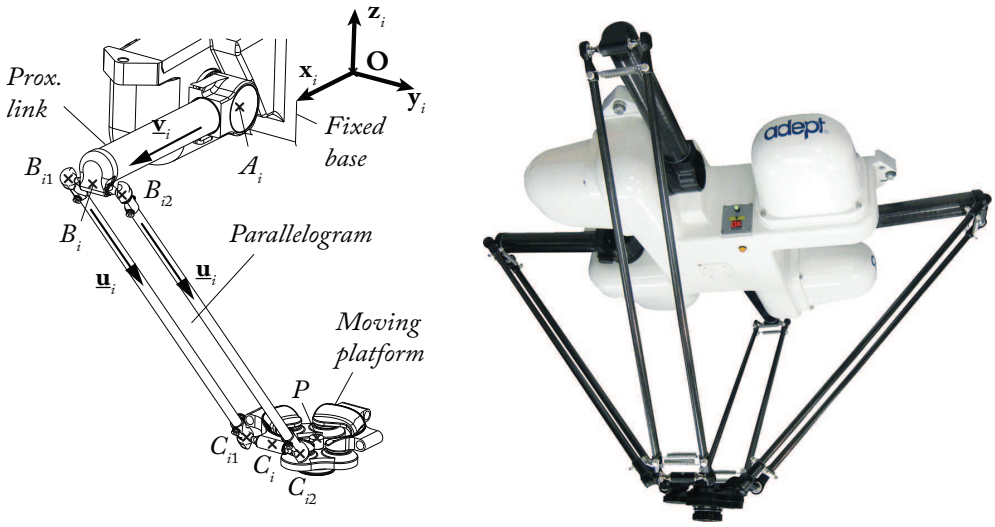
Singular configurations. The singular configurations of 3–$\underline{U}PS$-like robots have been deeply studied in the past [34, 35]. Type 2 singularities appear when the planes $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ (whose normal directions are defined by the vectors $\underline{\mathbf{u}}_1$, $\underline{\mathbf{u}}_2$ and $\underline{\mathbf{u}}_3$, respectively) and the plane $\mathcal{P}_4$ (passing through the points B_1, B_2 and B_3) intersect in one point (that can be at infinity) (Fig. 10(b)).

4.2.2. The Delta-like robot family

The Delta-like robot family regroups the robots made of n $\underline{A}$–{2–US} legs (where $\underline{A}$ can be either an active $\underline{R}$ or $\underline{P}$ joint – see Fig. 12(a) for an example of $\underline{R}$–{2–US} leg) [38, 44], or some of their variations such as $\underline{A}UU$ [45], or even $\underline{A}US$ legs [46, 47], i.e. the leg is composed of one active $\underline{R}$ or $\underline{P}$ joint located on the ground, followed by either two passive U joints, a spatial parallelogram (2–US loops) or even passive U and S joints.

Probably the most known robots of this family are the Delta [38], the Quattro [39] (Fig. 12(b)), the Orthoglide [40], but many other types of architectures exist (see for example [44, 48]). For such types of legs, the distal links direction can be observed (Fig. 12(a)). From Section 3.2, it can be shown that the virtual equivalent legs are:

- for $\underline{R}$–{2–US} legs: a Π–{2–$\underline{U}S$} or a Π–{2–$\underline{U}U$} leg (Fig. 13(a));



(a) a $\underline{R}$–{2–$\underline{US}$} leg (b) the Adept Quattro

Figure 12: Example of leg and of robot of the Delta-like family

- for $\underline{R}UU$ legs: a $\Pi\underline{U}U$ leg;
- for $\underline{R}US$ legs: a $\Pi\underline{U}S$ leg;
- for $\underline{P}$–{2–US} legs: a P–{2–$\underline{U}S$} or a P–{2–$\underline{U}U$} leg;
- for $\underline{P}UU$ legs: a $P\underline{U}U$ leg;
- for $\underline{P}US$ legs: a $P\underline{U}S$ leg;

i.e. the active R joints fixed on the ground are replaced by a passive planar parallelogram joint, while the active P joints becomes passive and the passive U or $R\Pi$ joints become active. Then, using the usual methodology, the *fkp* and singularity analysis can be carried out.

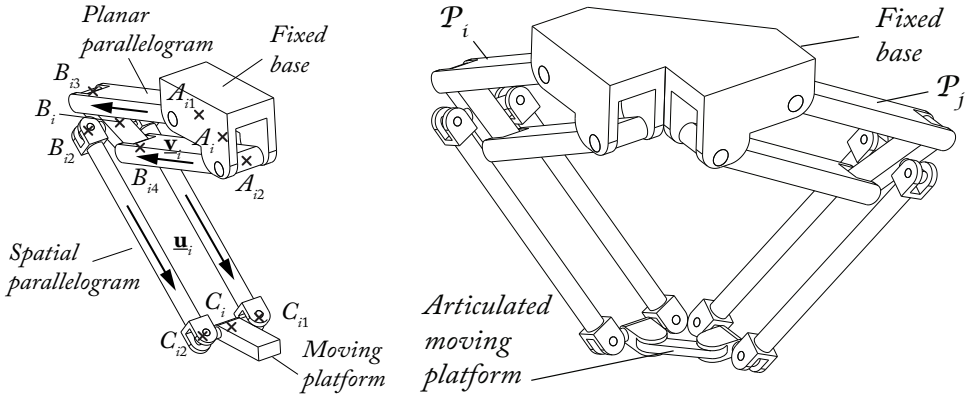
For illustrating this Section, let us present the *fkp* and singularity analysis of the hidden robot model of the Quattro with 4 *dof* (that can perform Schoenflies motions), when controlled using leg direction observation.

The Quattro is made of 4 $\underline{R}$–{2–US} legs, thus its equivalent hidden robot will be made of Π–{2–$\underline{U}S$} or Π–{2–$\underline{U}U$} legs. As such hidden robot legs have 2 degrees of actuation (the U joint is fully actuated), only two legs have to be observed for fully controlling the Quattro using leg direction observation. However in this case, if the hidden robot has a 2–Π–{2–$\underline{U}S$} architecture, the platform will have two uncontrolled *dof*. This phenomenon disappear if Π–{2–$\underline{U}U$} legs are used in the hidden robot model (Fig. 13 – in this picture, the articulated platform is simplified for a clearer drawing, but has indeed the kinematic architecture presented in Fig. 14).

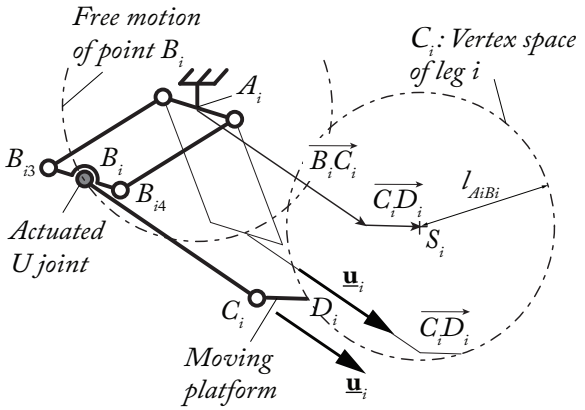
Forward kinematics and assembly modes. Without loss of generality, let us consider that we analyze the 2–Π–{2–$\underline{U}U$} robot depicted at Fig. 13(a). Looking at the vertex space of each leg when the active $\underline{U}$ joints are fixed, the points C_i and D_i are carrying out a circle $\mathcal{C}_i$ of radius $l_{A_iB_i}$ centred in S_i (Fig. 13(c)).

The Quattro with 4 *dof*, and consequently its hidden robot model, has a particularity: its platform is passively articulated (Fig. 14) so that its orientation with respect to the horizontal plan xOy stays constant, while it can have one degree of rotation around the z axis, i.e. point D_2 can describe a circle $\mathcal{C}_l$ located in the horizontal plane, centred in D_1 and with a radius $l_{D_1D_2}$. For solving the forward kinematics, it is thus necessary to virtually cut the platform at point D_2 and to compute the coupler surface of point D_2 when it belongs to leg 1. This coupler surface is the surface generated by $\mathcal{C}_l$ when it performs a circular translation along $\mathcal{C}_1$. Such a surface is depicted in Fig. 15(a) and is called a Bohemian Dome [49].

A Bohemian Dome is a quartic surface, i.e. an algebraic surface of degree 4. When it intersects the vertical plane $\mathcal{P}_l$ containing the circle $\mathcal{C}_2$ (i.e. vertex space of the second leg), the obtained curve is a quartic

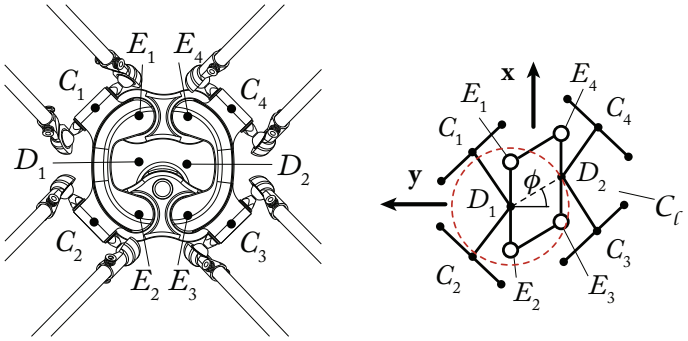


(a) a Π–{2–$\underline{U}U$} leg (b) the hidden robot model for the Adept Quattro



(c) vertex space of point D_i (projection of the Π–{2–$\underline{U}U$} leg in a vertical plane)

Figure 13: Example of leg and of hidden robot for the Delta-like family

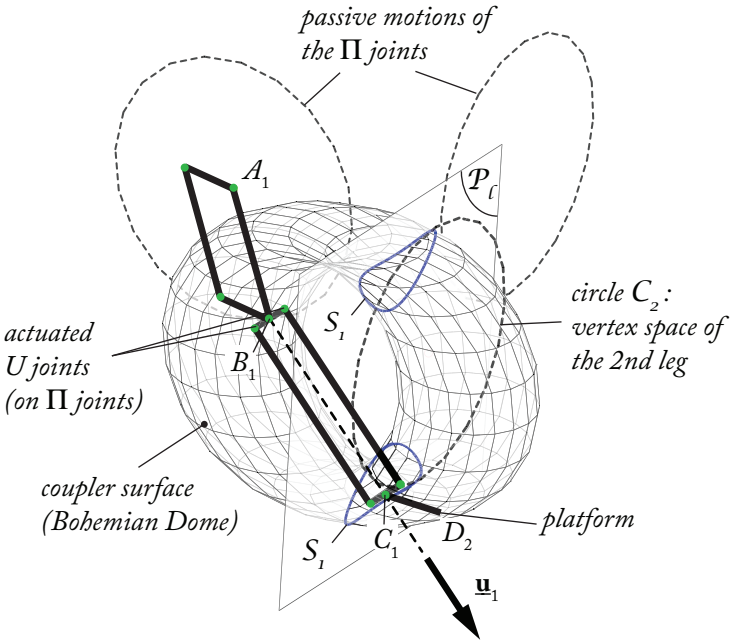


(a) CAD model (b) Schematics

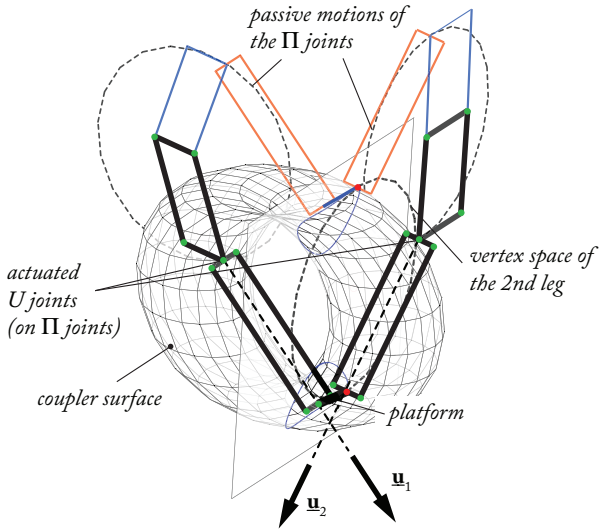
Figure 14: The platform of the Quattro

curve (denoted at $\mathcal{S}_1$ – Fig.15(a)). And using the Bézout theorem [50], it can be proven that, when the circle corresponding to the vertex space of leg 2 intersects this quartic curve, there can exist at most 8 intersection points, i.e. 8 assembly modes. Some examples of assembly modes for the 2–Π–{2–$\underline{U}U$} robot are depicted in Figs. 15(b) and 15(c).

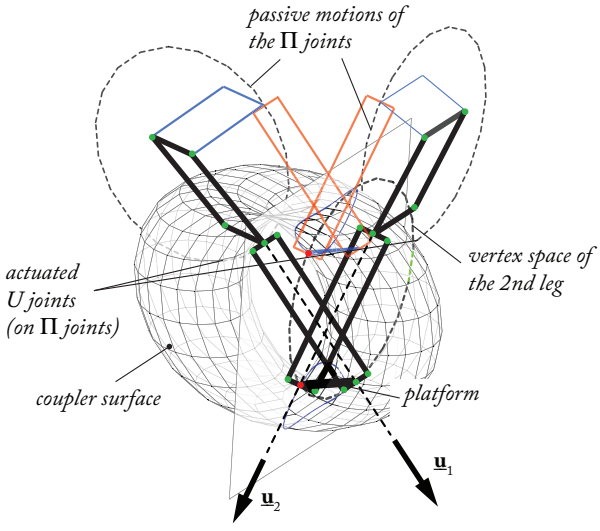
It should be noted that, when circles C_1 and C_2 are located in parallel planes, $\mathcal{S}_1$ degenerates into 1 or 2 circles. In this case, the maximal number of assembly modes decreases to 4. It must be mentioned here that, in usual controllers when only the encoder data is used, the number of assembly modes of the Quattro is equal to 8.



(a) Coupler surface when leg 2 is disconnected



(b) First set of possible assembly modes



(c) Second set of possible assembly modes

Figure 15: Solutions of the *fkp* for a 2–Π–{2–$\underline{U}U$} robot (in this example, only 4 assembly modes exist)

Singular configurations. For the 2–Π–{2–$\underline{U}U$} robot, Type 2 singularities appear when the planes $\mathcal{P}_i$ and $\mathcal{P}_j$ (whose normal vectors are equal to $\underline{\mathbf{v}}_i^{\perp}$ and $\underline{\mathbf{v}}_j^{\perp}$, resp.) are parallel. In such cases, the circle C_2 is tangent to the Bohemian Dome at their intersection point and the robot gains one uncontrollable *dof* along this tangent (Fig. 16).

4.3. Discussion

Thus, the concept of hidden robot model, via the use of tools for the geometric and kinematic analysis of parallel robots developed by the mechanical design community, can really help the control community to simplify the verification of the controllability (*and above all, to certify the results*) of parallel robots controlled using leg-direction-based visual servos. With such an approach,

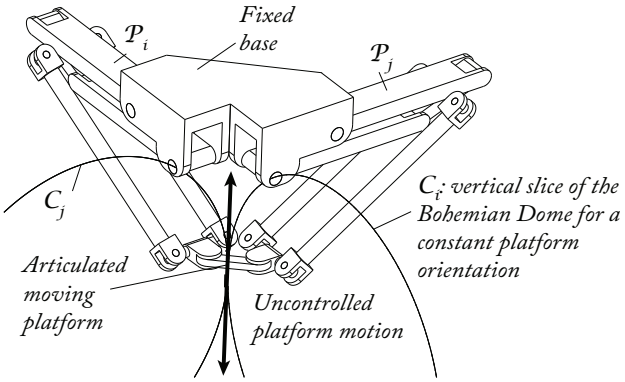
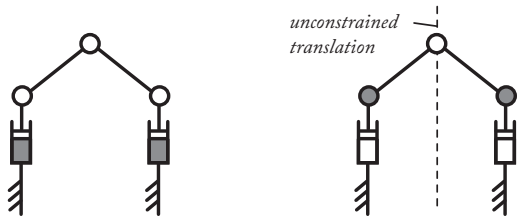


Figure 16: Example of a Type 2 singularity for a 2–Π(2–$\underline{U}U$) robot: the platform gets an uncontrollable translation.



(a) The $\underline{P}RRR\underline{P}$ robot

(b) Its hidden robot model: a $P\underline{R}\underline{R}RP$ robot

Figure 17: The $\underline{P}RRR\underline{P}$ robot and its hidden robot model

- the problem of the correct convergence of the robot to the desired pose can be analyzed (i.e. *it was possible to analyze if a global diffeomorphism between the Cartesian space and the leg direction space exists*),
- as well as the problem of the singularity analysis of the mapping between the leg direction space and the Cartesian space (*including the problem of finding local minima to which the robot could converge*),

by reducing the problem to the kinematic analysis of another robot (in all the studied cases, it was possible to find simple geometric conditions without any mathematical derivations for the *fkp* and the singular configurations).

It is interesting to notice that, for most of the studied robots, the hidden robots models have less assembly modes, i.e the proposed control approach based on the observation of the leg directions allows the simplification of the *fkp* for the studied architectures. This is obvious not a general assertion, but this appears for most of robots we studied.

Considering the singularity analysis of the mapping between the leg direction space and the Cartesian space, typical examples for which the verification of the controllability is easy to carry out without any mathematical derivations using the concept of hidden robot model are the cases of $\underline{P}RRR\underline{P}$ robots with all $\underline{P}$ parallel (Fig. 17(a)) and of Delta-like robots actuated via $\underline{P}$ joints for which all $\underline{P}$ are parallel (such as the UraneSX or the I4L [44, 48]). It was shown in [17] through the analysis of the rank deficiency of the interaction matrix that it was not possible to control such types of robots using leg direction observation. Considering this problem with the hidden robot concept is very easy. For example, in the case of the $\underline{P}RRR\underline{P}$ robot with parallel $\underline{P}$ joints, the hidden robot has a $P\underline{R}\underline{R}RP$ architecture (Fig. 17(b)), where the parallel P joints are passive. This robot is well-known to be architecturally singular as there is no way to control the translation along the axis of the parallel P joints. This result can be easily extended to the cases of the hidden robots of the UraneSX and the I4L. More developments on the controllability analysis of parallel robots controlled using leg-direction-based visual servos can be found in [51].

It should finally be mentioned that in order to avoid the loss of controllability of such robot architectures, a way to proceed is to modify the controller so that the second component $\underline{\mathbf{h}}_i$ of the Bi-normalised Plücker

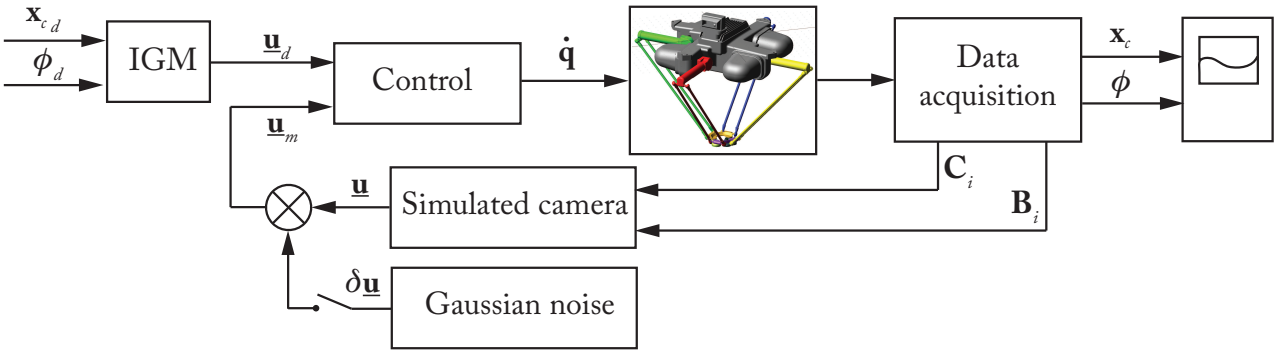


Figure 18: The controller used for the simulations

coordinates of the lines is considered. In such a case, other hidden robots should be found, but this is part of our future work.

5. Case Study

In this Section, simulations and experiments are performed on the Adept Quattro presented in Section 4.2.2.

5.1. Simulation results

The geometric parameters of the Quattro are given in Appendix Appendix B, as well as the equations necessary to perform the following simulations.

5.1.1. Description of the simulator

In this section, simulations are performed on an ADAMS mockup of the Adept Quattro (Fig. 18) with the same kinematic properties as the real robot by Adept. This virtual mockup is connected to Matlab/Simulink via the module ADAMS/Controls. The controller presented in Section 2.3 is applied (with $\lambda = 0.8$ – Fig. 18) in which:

- the observation of the leg is simulated by extracting in the ADAMS model the positions of the anchor points of each parallelograms,
- the actuated joint velocities are given as inputs of the ADAMS mockup.

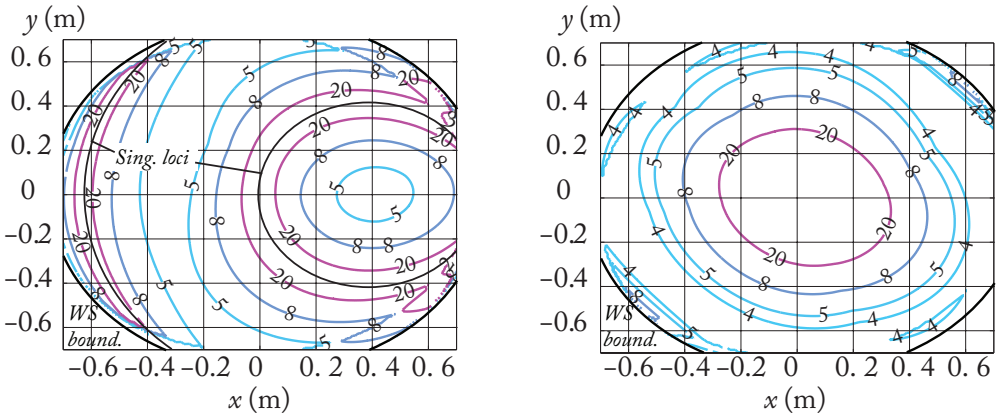
On the used control scheme, we can also switch on or off simulated measurement noise (whose amplitude can be parameterized) on the observation of the leg directions.

5.1.2. Accuracy analysis of the Quattro using leg observation

First, let us make the analysis of the accuracy of the Quattro using leg observation based on the accuracy model depicted in Appendix Appendix A.1. For this mechanism, in the case of a leg direction based visual servoing and for an error ${}^b\delta\underline{\mathbf{u}}_i$ defined such that the vector ${}^b\underline{\mathbf{u}}_i$ is contained in a cone of axis ${}^b\underline{\mathbf{u}}_{i0}$ and of half angle ψ_i (${}^b\underline{\mathbf{u}}_{i0}$ is the nominal value of ${}^b\underline{\mathbf{u}}_i$ and, in what follows, ψ_i is taken equal to 0.1 deg for each leg direction), let us first compute the maximal positioning and orientation error when only two of its four legs are observed. Six different combinations are possible. However, the value of the error for only two of them (when legs $\{2, 3\}$ and $\{2, 4\}$ are observed) is plotted at Figs. 19 and 20.

In Figs. 19(a) and 20(a), it is possible to note that the maximal error varies very quickly, especially near singularities of the hidden robot. In Figs. 19(b) and 20(b), things are different. The variation of the accuracy is smoother for the orientation error, and the position accuracy decrease in the middle of the workspace only. Thus, it can be concluded that the selection of the legs to observe is crucial for the final pose accuracy.

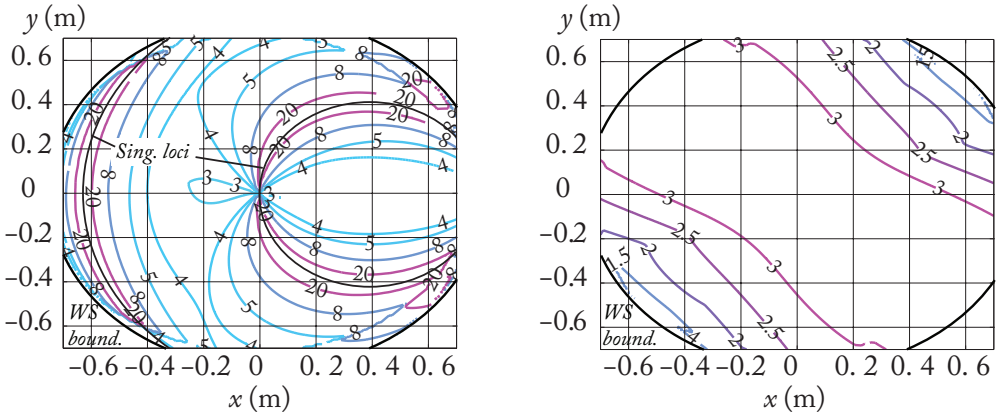
Let us now compute the maximal positioning error when the four legs are observed. It can be observed that the position error is larger near $\{x = 0m, y = 0m, z = -0.61m, \phi = 0deg\}$. This can be explained by the



(a) legs {2, 3} are observed

(b) legs {2, 4} are observed

Figure 19: Maximal position error (in mm) for $z = -0.7$ m and $\phi = 0$ deg. Singularity loci are singularity cases of the hidden robot associated with the leg-direction-based visual servoing of the Quattro.



(a) legs {2, 3} are observed

(b) legs {2, 4} are observed

Figure 20: Maximal orientation error (in deg) for $z = -0.7$ m and $\phi = 0$ deg. Singularity loci are singularity cases of the hidden robot associated with the leg-direction-based visual servoing of the Quattro.

fact that this configuration is a singularity of the model for which all the planes $\mathcal{P}_i$ ($i = 1, 2, 3, 4$) are parallel. Thus, even if all the legs are observed, singular configurations may appear near which the accuracy is poor. Such a phenomenon shows the importance of the study of the intrinsic properties of the controller via the hidden robot concept.

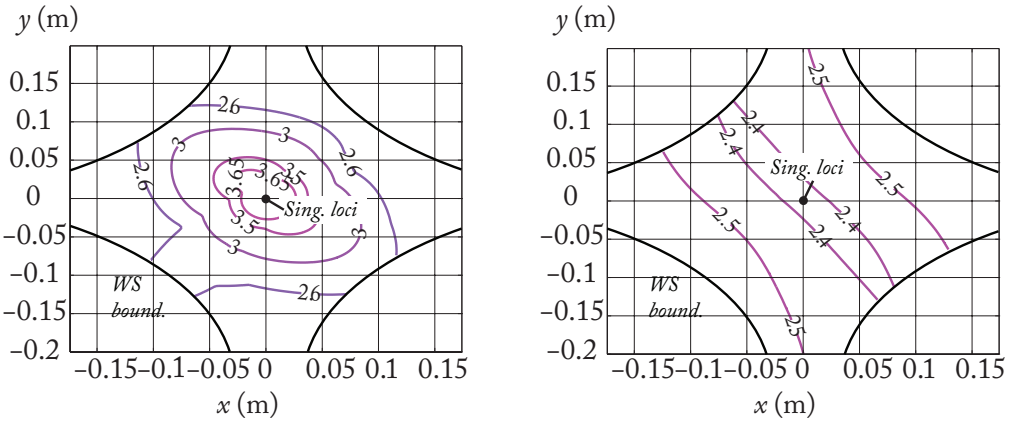
These results will be compared with the results obtained in the following numerical simulations.

5.1.3. Numerical validations

Testing the convergence of the robot to the desired pose

In the first simulation, no noise is added on the simulated values of the leg directions. The initial platform pose is equal to $\{x = 0m, y = 0m, z = -0.75m, \phi = 0deg\}$ and the final desired platform pose is set to $\{x = -0.2m, y = 0m, z = -0.56m, \phi = 0deg\}$. For going from the initial point to the final ones, two sets of observed leg directions are tested: $\{1, 4\}$ and $\{2, 3\}$. For those two set of legs, solving the *fkp* of the hidden robot model of the Quattro presented in Section 4.2.2 at the desired final configuration of the robot, the following assembly modes can be obtained:

- for legs $\{1, 4\}$:
  - solution 1: $\{x = -0.2m, y = 0m, z = -0.56m, \phi = 0deg\}$



(a) position error (mm)

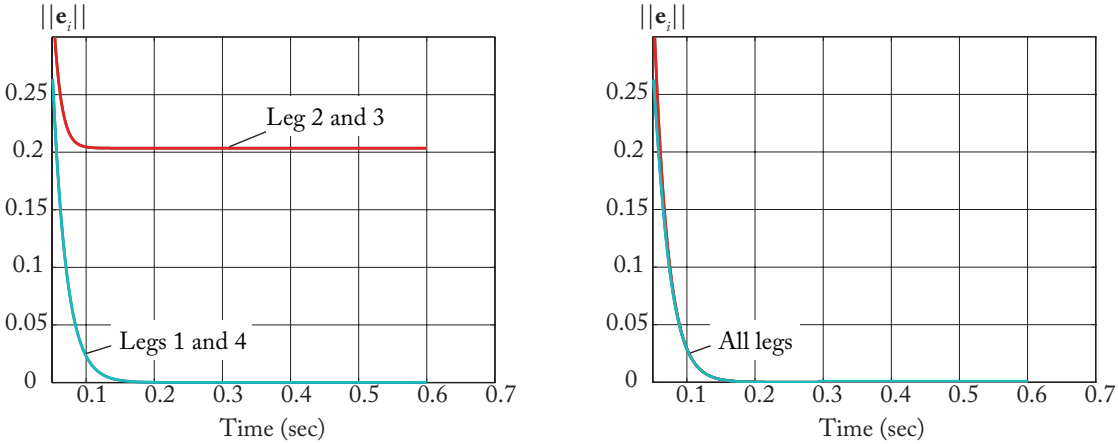
(b) orientation error (deg)

Figure 21: Maximal position and orientation error for $z = -0.61$ m and $\phi = 0$ deg when all legs are observed. Singularity loci are singularity cases of the hidden robot associated with the leg-direction-based visual servoing of the Quattro.

- solution 2: $\{x = -0.2m, y = 0m, z = -0.909m, \phi = 0deg\}$
- solution 3: $\{x = -0.138m, y = 0.062m, z = -1.019m, \phi = 0deg\}$
- solution 4: $\{x = -0.138m, y = 0.062m, z = -0.45m, \phi = 0deg\}$

- for legs $\{2, 3\}$:
  - solution 1: $\{x = -0.2m, y = 0m, z = -0.56m, \phi = 0deg\}$
  - solution 2: $\{x = -0.2m, y = 0m, z = -0.296m, \phi = 0deg\}$
  - solution 3: $\{x = -0.262m, y = 0.062m, z = -0.694m, \phi = 0deg\}$
  - solution 4: $\{x = -0.262m, y = 0.062m, z = -0.161m, \phi = 0deg\}$

The results for the convergence of the leg directions are presented in Fig. 22. It can be shown that when the legs $\{2, 3\}$ are observed, all leg directions converge to 0. This is not true for the second case. Looking at the platform pose computed by ADAMS, the robot reach the configuration $\{x = -0.2m, y = m, z = -0.909m, \phi = 0deg\}$, i.e. the second solution for the *fkp* of the hidden robot model of the Quattro (Fig. 23).



(a) when legs $\{1, 4\}$ are observed

(b) when legs $\{2, 3\}$ are observed

Figure 22: Error norm on each leg $\|\mathbf{e}_i\|$.

A second simulation is performed in which all legs are observed. The initial platform pose is equal to $\{x = 0.05m, y = 0.05m, z = -0.8m, \phi = 0deg\}$ and the final desired platform pose is set to $\{x = 0.03m, y =$

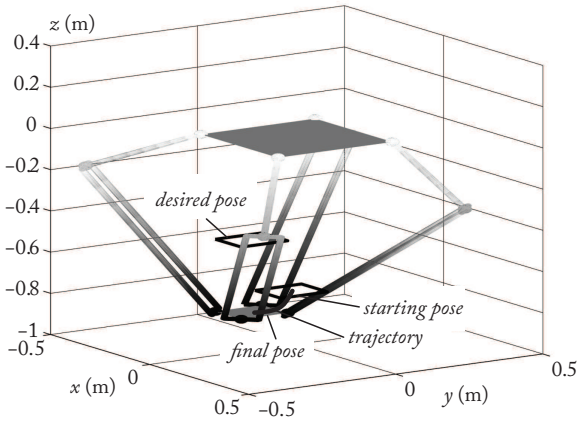


Figure 23: Initial, desired and final position when legs $\{1, 4\}$ are observed

$0.03m, z = -0.59m, \phi = 0deg\}$. Solving the *fkp* of the hidden robot model of the Quattro when all legs are observed at the desired final configuration of the robot, it can be proven that there still exist two assembly modes, separated by singularities (Fig. 21) which are:

- solution 1: $\{x = 0.03m, y = 0.03m, z = -0.59m, \phi = 0deg\}$
- solution 2: $\{x = 0.03m, y = 0.03m, z = -0.65m, \phi = 0deg\}$

Looking at the platform pose computed by ADAMS, even if all errors on the legs vanish (Fig. 24), the robot reaches the configuration $\{x = 0.03m, y = 0.03m, z = -0.65m, \phi = 0deg\}$, i.e. the second solution for the *fkp* (Fig. 25).

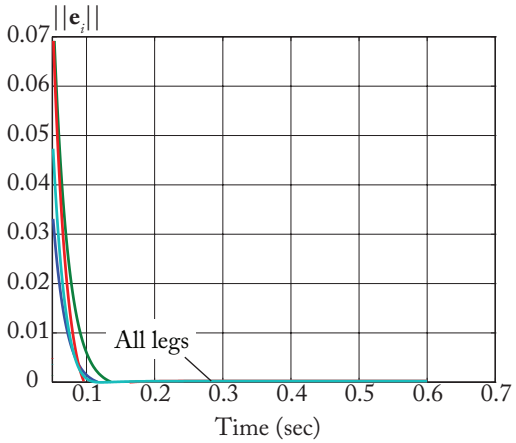


Figure 24: Error norm on each leg $\|\mathbf{e}_i\|$ when all the legs are observed.

All these numerical results confirm the presence of the virtual robot hidden within the controller that must be studied in order to avoid convergence problems due to inadequate stacking of interaction matrices.

Testing the presence of local minima

For that simulations, all legs are observed. The initial platform pose is equal to $\{x = 0.028m, y = 0m, z = -0.617m, \phi = 0deg\}$ and the final desired platform pose is set to $\{x = -0.6m, y = 0m, z = -0.8m, \phi = 0deg\}$. No noise is added on the simulated values of the leg directions. After about 0.3 s of simulations, the robot stops in the configuration $\{x = -0.588m, y = 0m, z = -0.847m, \phi = 0deg\}$ while the error on the leg direction is far from zero (Fig. 26). Thus we are in the presence of a local minimum.

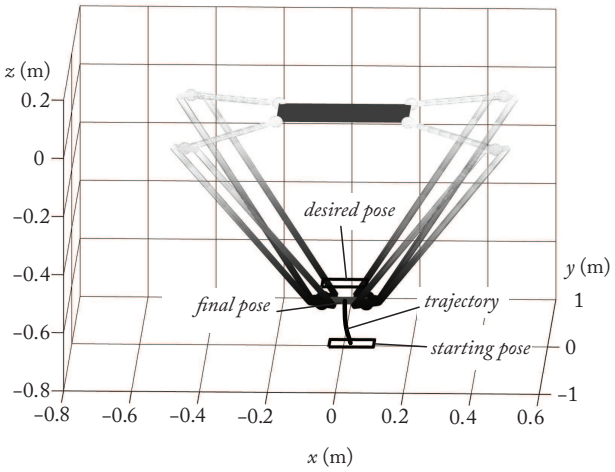
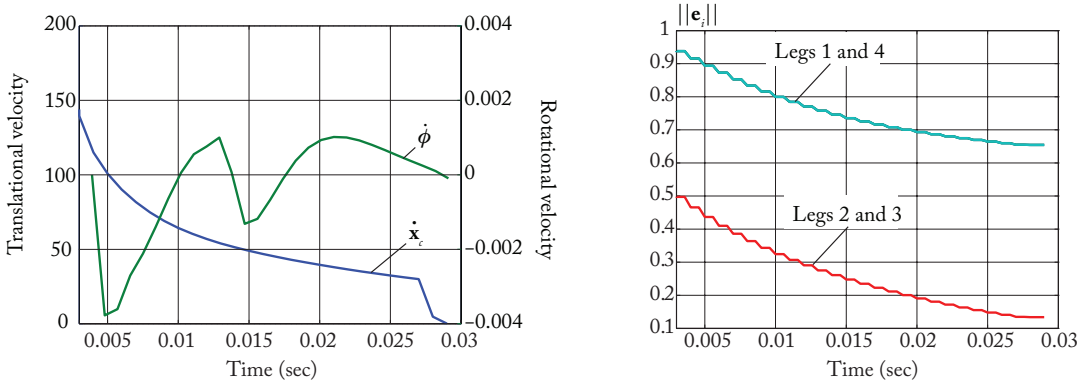


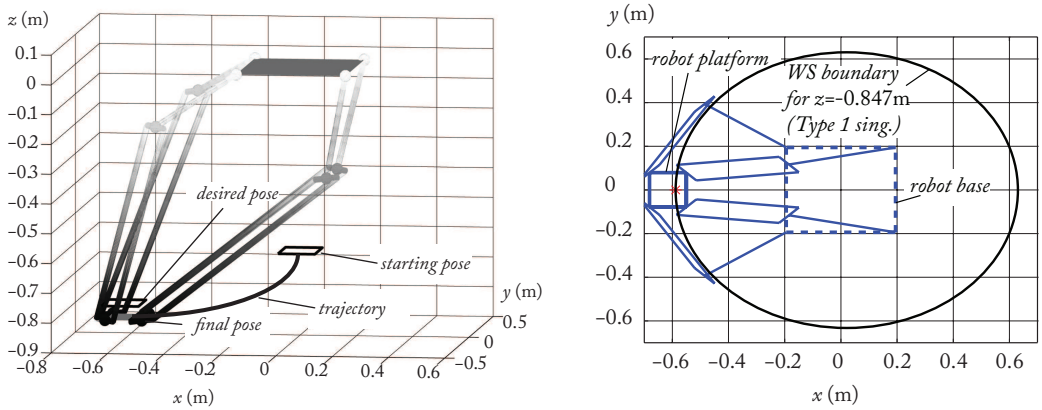
Figure 25: Initial, desired and final position when all legs are observed



(a) platform velocity

(b) error norm on each leg $||\mathbf{e}_i||$

Figure 26: Platform velocity and error norm on each leg when the robot meets a local minimum.



(a) 3D view

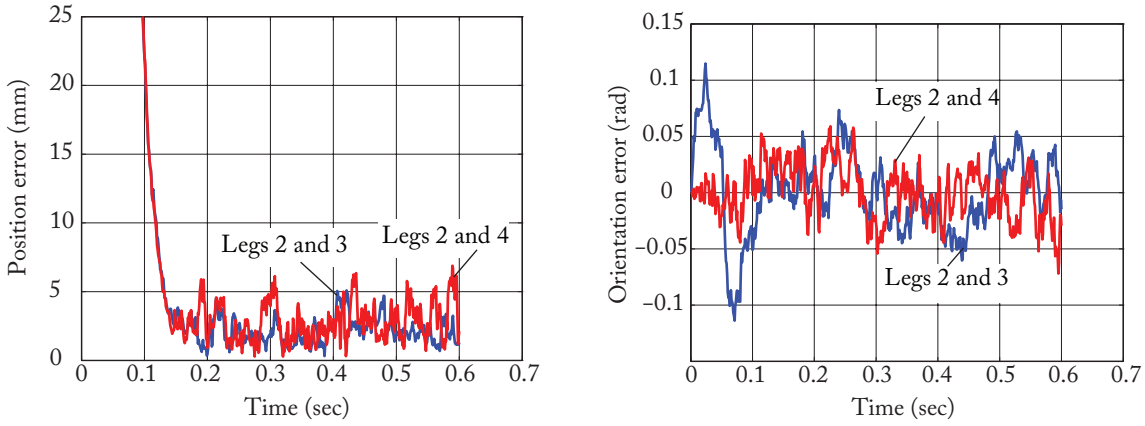
(b) top view

Figure 27: Robot configuration when it meets a local minimum.

Looking at the configuration in which the robot is blocked, it appears that, as forecasted, it is a Type 1 singularity (boundary of the workspace (Fig. 27)). This confirms the fact that the local minima appear in the Type 1 singularities of the hidden robot model, as mentioned in Section 3.3.

Testing the importance of the selection of the observed legs on the robot accuracy

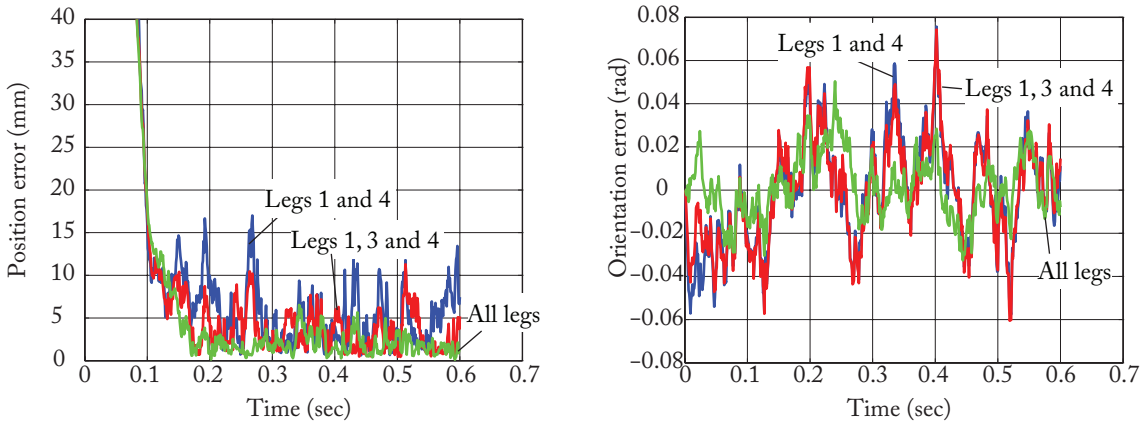
In the first simulation, the initial platform pose is equal to $\{x = 0.02m, y = 0.1m, z = -0.7m, \phi = 0deg\}$ and the final desired platform pose is set to $\{x = -0.2m, y = 0.01m, z = -0.7m, \phi = 0deg\}$. A random noise of 0.1 deg is added to the simulated measure of the leg directions. To show the importance of the leg selection on the robot accuracy, it is decided to control the robot displacement using two different sets of legs: (i) legs $\{2, 3\}$ and (ii) legs $\{2, 4\}$. The results (Fig. 28) show that, as presented in Fig. 19, the final platform pose accuracy is better when legs $\{2, 3\}$ are observed (around 3 mm and 0.05 rad) than with legs $\{2, 4\}$ (around 7 mm and 0.07 rad).



(a) position error (mm) (b) orientation error (rad)

Figure 28: Orientation and position error when legs $\{2, 3\}$ and legs $\{2, 4\}$ are observed.

In the second simulation, the initial platform pose is equal to $\{x = 0.05m, y = 0.05m, z = -0.8m, \phi = 0deg\}$ and the final desired platform pose is set to $\{x = 0.03m, y = 0.03m, z = -0.65m, \phi = 0deg\}$. It is decided to control the robot displacement using three different sets of legs: (i) legs $\{1, 4\}$, (i) legs $\{1, 3, 4\}$ and (iii) all legs. The results (Fig. 29) show that the final platform pose accuracy is better when all legs are observed, while the accuracy is quite the same when two or three legs are observed. However, this result must not hide the fact that, even if four legs can lead to better accuracy, some convergence problems can still appear, as shown previously.



(a) position error (mm) (b) orientation error (rad)

Figure 29: Orientation and position error when legs $\{1, 4\}$, $\{1, 3, 4\}$ and all legs are observed.

All these simulations validate the theory presented in Section 3.

5.2. Experimental results

5.2.1. Description of the benchmark

In this section, experiments are now performed on a real Adept Quattro. The benchmark is composed of (Fig. 30):

- an Adept Quattro robot bought by the Institut Pascal of Clermont-Ferrand (France),
- a camera AVT Marlin F131B firewire IEEE1394 (lens: 3.6mm 1:1.6 1/2 inch for CCD camera), which is mounted at the centre of the robot base so that all the legs can be observed without any problems of occlusion and whose intrinsic and extrinsic parameters have been calibrated,
- a lighting system that provides an homogenous lighting to the scene,
- a computer that extracts the data coming from the camera, computes the value of the leg directions $\underline{\mathbf{u}}$, then calculates the robot actuator velocity $\dot{\mathbf{q}}$ using the controller of Section 2.3.2 and send the information to the robot controller. Note that, in experiments, the value of λ in the controller is fixed to 0.2.

Moreover, the robot is covered by a cloth that prevents the lighting variations and guarantees the contrast quality required for observing the black legs of the robot (Fig. 31).

Finally, it must be mentioned we have deliberately decided to use the minimal camera resolution and to not undistort the image captured. The measurement noise on the leg direction is thus of about 0.1 rad, but:

- such a high noise is interesting to show the controller robustness to leg direction prediction errors,
- the noise is so high that, for analyzing the robot accuracy and measuring the distance between the real and nominal robot configurations, we can directly record and use the value of the platform pose predicted by the Adept Quattro controller instead of using one external measurement device (such as a lasertracker).

5.2.2. Experimental validations

Testing the convergence of the robot to the desired pose

We replay now experimentally the convergence tests presented in Section 5.1.3. The starting and desired final points are the same as previously. The results are presented in the Tables 2 to 4 and illustrated by the Figs. 32 to 34. It should be mentioned that, for cross-validating the results on those pictures, the plotted values of the error norms are computed using the values of the leg directions given by the Quattro controller.

Due to the presence of high measurement noise, the robot can of course not converge to the final desired pose. Therefore, in these Tables, information on the tolerable maximal error on the pose attained attained in simulations is given. Please note that, due to the large value of the error on the measured angle, the model defined in Section Appendix A.1 is no longer valuable and we have preferred to use a more refined non linearized model proposed in [52].

Table 2: Results on the experiments carried out for testing the convergence of the robot when legs 1 and 4 are observed (the positions are in meter, the angles in radians).

Desired final pose	$\{x = -0.2, y = 0, z = -0.56, \phi = 0\}$
Final pose in simulation	$\{x = -0.2, y = 0, z = -0.91, \phi = 0\}$
Tolerable position error	0.11 m
Tolerable orientation error	2.00 rad
Final pose in experiments	$\{x = -0.11, y = 0.01, z = -0.86, \phi = -2.15\}$
Distance to the final pose in simulation	0.10 m
Orient. err. w.r.t. the final pose in simulation	2.15 rad

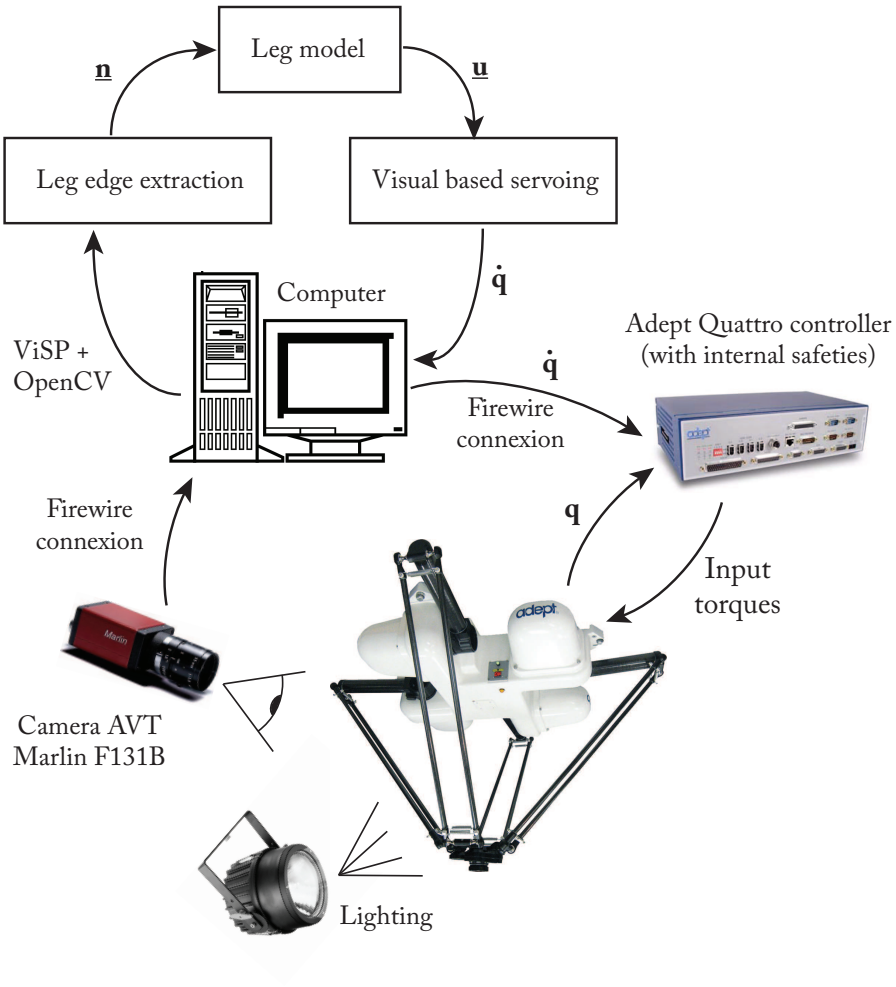


Figure 30: Experimental bench

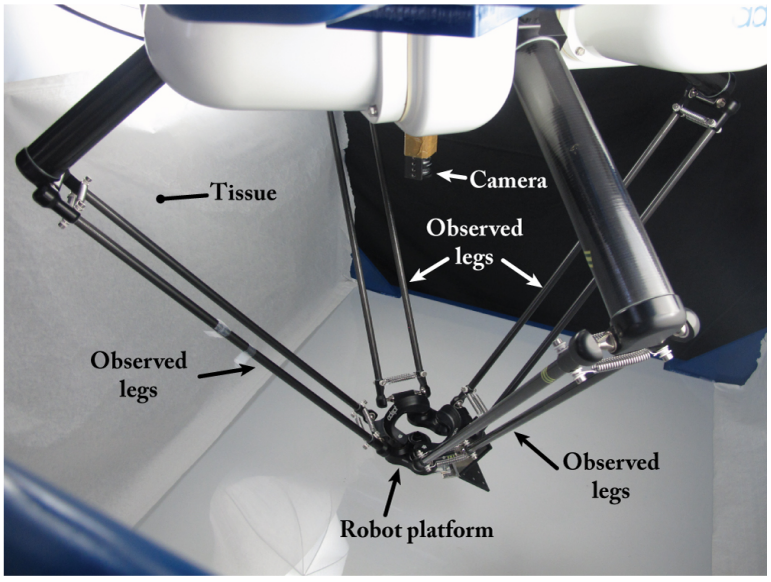
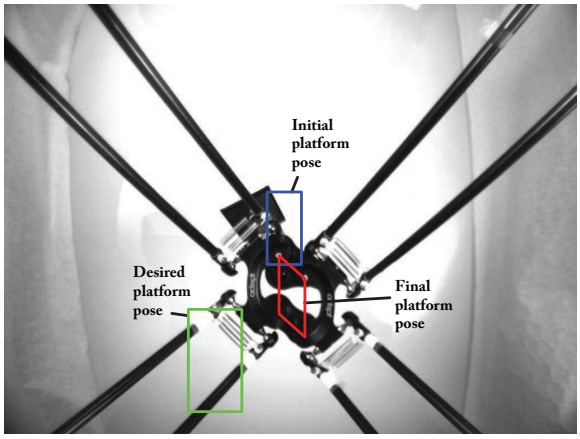
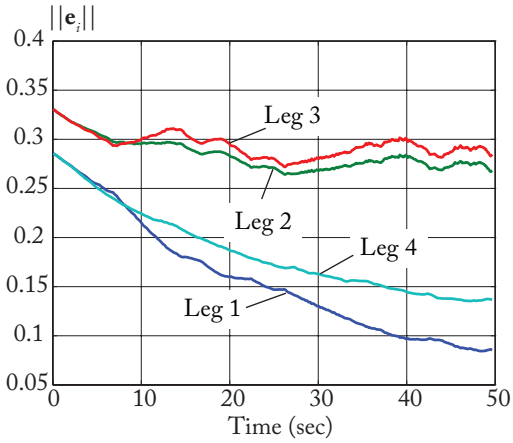


Figure 31: The Quattro recovered by the cloth

All these experimental results match with the simulation results presented above and confirm the presence of the virtual robot hidden within the controller that must be studied in order to avoid the convergence problems due to inadequate stacking of interaction matrices.

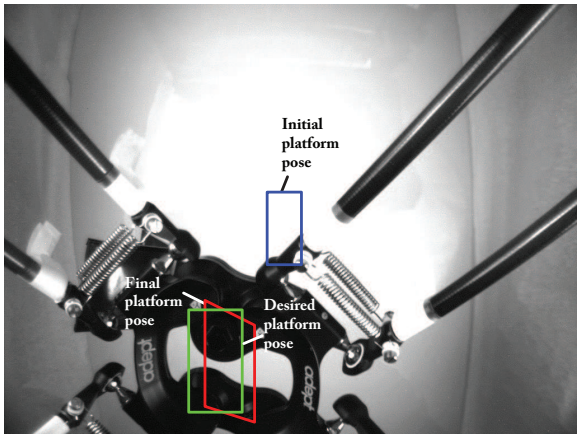


(a) top view of the platform

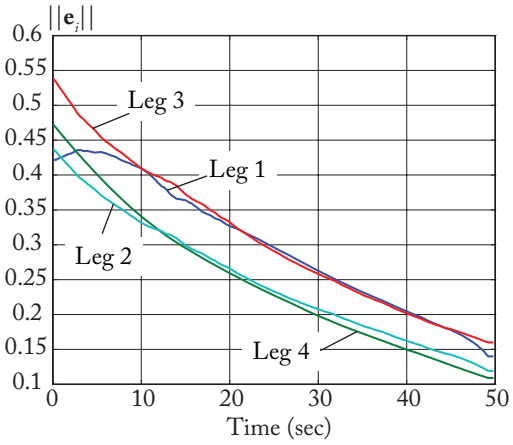


(b) error norm on each leg $\|\mathbf{e}_i\|$

Figure 32: Convergence of the robot when legs 1 and 4 are observed (desired pose: $\{x = -0.2, y = 0, z = -0.56, \phi = 0\}$).



(a) top view of the platform



(b) error norm on each leg $\|\mathbf{e}_i\|$

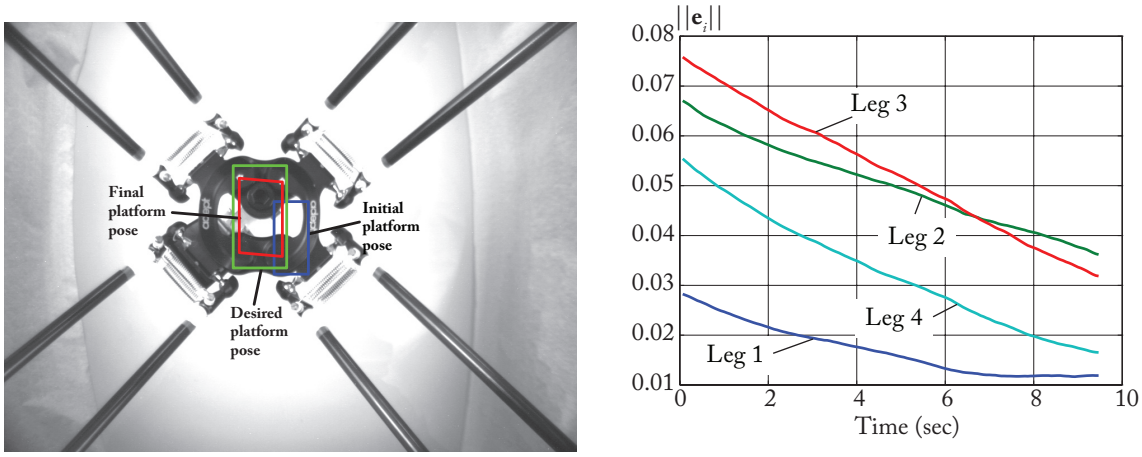
Figure 33: Convergence of the robot when legs 2 and 3 are observed (desired pose: $\{x = -0.2, y = 0, z = -0.56, \phi = 0\}$).

Table 3: Results on the experiments carried out for testing the convergence of the robot when legs 2 and 3 are observed (the positions are in meter, the angles in radians).

Desired final pose	$\{x = -0.2, y = 0, z = -0.56, \phi = 0\}$
Final pose in simulation	$\{x = -0.2, y = 0, z = -0.56, \phi = 0\}$
Tolerable position error	0.23 m
Tolerable orientation error	1.23 rad
Final pose in experiments	$\{x = -0.12, y = 0.05, z = -0.55, \phi = -0.90\}$
Distance to the final pose in simulation	0.10 m
Orient. err. w.r.t. the final pose in simulation	0.90 rad

Testing the presence of local minima

Unfortunately, we were not able to do such experiments as the robot controller is designed with safeties that cannot be suppressed and that prevent going into singularities. However, as the presence of local minima that are located in the Type 1 singularities was demonstrated in simulations, we think that this numerical proof brings enough strength to our demonstration concerning this point.



(a) top view of the platform

(b) error norm on each leg $\|\mathbf{e}_i\|$

Figure 34: Convergence of the robot when all legs are observed (desired pose: $\{x = 0.03, y = 0.03, z = -0.59, \phi = 0\}$).

Table 4: Results on the experiments carried out for testing the convergence of the robot all legs are observed (the positions are in meter, the angles in radians).

Desired final pose	$\{x = 0.03, y = 0.03, z = -0.59, \phi = 0\}$
Final pose in simulation	$\{x = 0.03, y = 0.03, z = -0.65, \phi = 0\}$
Tolerable position error	0.08 m
Tolerable orientation error	1.54 rad
Final pose in experiments	$\{x = 0.05, y = 0.03, z = -0.72, \phi = 0.05\}$
Distance to the final pose in simulation	0.07 m
Orient. err. w.r.t. the final pose in simulation	0.05 rad

Testing the importance of the selection of the observed legs on the robot accuracy

We replay now experimentally the accuracy tests presented in Section 5.1.3. The starting and desired final points are the same as previously, as well as the observed legs. Each experiment is run five times and we present here the maximal values obtained on the position and orientation error. The results are shown in the Tables 5 to 6.

Table 5: Results on the experiments carried out for testing the accuracy of the robot when legs $\{2, 3\}$ or $\{2, 4\}$ are observed.

Desired final pose	$\{x = -0.2m, y = 0.01m, z = -0.7m, \phi = 0deg\}$	
Legs	$\{2, 3\}$	$\{2, 4\}$
Position error	0.11 m	0.23 m
Orientation error	0.06 rad	0.68 rad

Table 6: Results on the experiments carried out for testing the accuracy of the robot when legs $\{1, 4\}$, $\{1, 3, 4\}$ or $\{1, 2, 3, 4\}$ are observed.

Desired final pose	$\{x = 0.03m, y = 0.03m, z = -0.65m, \phi = 0deg\}$		
Legs	$\{1, 4\}$	$\{1, 3, 4\}$	$\{1, 2, 3, 4\}$
Position error	0.11 m	0.09 m	0.07 m
Orientation error	0.39 rad	0.31 rad	0.05 rad

Once again, all these experimental results match with the simulation results presented above and confirm the necessity to carefully select the set of legs to observe in order to obtain the best accuracy possible. However, it must be recalled that, even if observing all the legs lead to a better accuracy, this result must not hide

the fact that some convergence problems can still appear, as shown previously.

All these experiments validate the theory presented in Section 3.

5.3. Discussions and future works

All the results presented above show the validity of the approach and also its importance: stacking several interaction matrices to derive a control scheme without doing a deep analysis of the intrinsic properties of the controller is clearly not enough. For avoiding the singularity problem due to the mapping between the robot space and leg space, whatever the number of observed legs (as, even if all legs are observed, there may be singularities of the mapping), the hidden robot kinematics must be analyzed to avoid the convergence and inaccuracy problems.

We would also like to add that, in this paper, we have deliberately chosen not to define the controller in the image space. There exist several reasons which justify that choice:

- the proof that the robot can be controlled using only the observation of the leg edges has been given in [15] and gives similar results as for the control with the leg directions, except that the robot accuracy was better. This can be explained by the fact that, as mentioned previously, the leg edges are only used as a measure of $\underline{\mathbf{u}}_i$. So *the problem is the same*, except in the fact that we must consider the singularity of the mapping between the edges and the leg directions, but this problem is well handled [15].

- in our benchmark, it can be shown that the variations of the vectors ${}^c\underline{\mathbf{n}}_i^j$ is very small in the whole workspace. The difference on vectors ${}^c\underline{\mathbf{n}}_i^j$ for two distinct robot configurations – whatever they are – is about 0.01 rad. So we must be sure that the measurement noise is much lower than this value by refining the data extraction from the image (e.g. by using some subpixellic approach) and this is not currently our main goal. Moreover, this small variation on the values of ${}^c\underline{\mathbf{n}}_i^j$ make us think that:

  - we should modify the controller so that the second component of the Bi-normalised Plücker coordinates of the lines corresponding to the edges is considered in order to get all the information contained in the image (direction and position of the lines). In such a case, other hidden robots should be found,
  - as the Quattro was not designed to be controlled using such a visual servoing approach, it may be unrealistic to try to obtain good results in terms of accuracy using vision. However, an interesting point would be, before the robot is designed, to take into account the control approach that will be used so that the leg positions (and as a result the robot architecture) can be optimized w.r.t. the desired visual servoing approach.

  These two points are parts of our future work.

It should also be mentioned that, in the present paper, we have deliberately focused on the study of the hidden robot models of several classes of parallel robots. This work is crucial in order to simplify the use of this tool by control engineers willing to develop of leg-observation-based controller. Others challenges are to use the hidden robot model tool:

- in order to propose a classification of the robots which cannot or can be controlled with leg-observation-based controllers and also.

- in order to help the control engineer to understand what types of information are missing in order to ensure the controllability of their robots.

Insights on these two last points can be found in [51].

6. Conclusions

This paper has presented a tool named the "Hidden robot concept" that is well addressed for analyzing the controllability of parallel robots in leg-observation-based visual servoing techniques. It has been shown that the mentioned visual servoing techniques involves the existence of a virtual robot model, hidden into the controller, that is different from the real controlled robot. Considering this hidden robot model allowed the finding of a minimal representation for the leg-observation-based control of the studied robots that is linked to a virtual hidden robot which is a tangible visualization of the mapping between the observation space and the real robot Cartesian space. It has been shown that the hidden robot model can be used to:

1. explain why the observed robot which is composed of n legs could be controlled in most cases using the observation of only m leg directions ($m < n$), knowing the fact that the minimal number of observed legs should be, for 3D unit vectors, an integer greater than $n/2$,
2. prove that there does not always exist a global diffeomorphism between the Cartesian space and the leg direction space,
3. simplify the singularity analysis of the mapping between the leg direction space and the Cartesian space by reducing the problem to the singularity analysis of a new robot,
4. certify that the robot will not converge to local minima, through the application of tools developed for the singularity analysis of robots.

A general way to find the find the hidden robot models corresponding to the real robot controlled via leg-observation-based visual servoing techniques has been shown and the hidden robot models of some well known classes of parallel robots have been studied. It has been proven that, using this concept, it is possible to demonstrate, using tools developed by the mechanical design community, that the robot can be controlled or not with the aforementioned visual servoing techniques.

Finally, numerical simulations and experimental validations made on an Adept Quattro robot have demonstrated the validity of the theoretical developments.

Thus, the concept of hidden robot model, associated with mathematical tools developed by the mechanical design community, is a powerful tool able to analyze the intrinsic properties of some controllers developed by the visual servoing community. Moreover, this concept showed that in some visual servoing approaches, stacking several interaction matrices to derive a control scheme without doing a deep analysis of the intrinsic properties of the controller is clearly not enough. Further investigations are required.

Appendix A. Selection of the controlled legs

Depending on the chosen interaction matrices, i.e. on the choice of the observed legs, the geometry of the hidden robot models will vary, as well as its singularities and assembly modes. As singularities divide the workspace into distinct aspects [2], it is necessary to study the motion feasibility by selecting a set of legs that can allow the robot displacement. Moreover, even if the motion is feasible, if the robot goes close to a singularity, the positioning error can considerably grow.

Therefore, it is necessary to find the best set of legs to observe in order to get the best performances of the robot w.r.t. a desired task. This is the main goal of this appendix.

Appendix A.1. Definition of criteria for selecting the legs to observe

Several indices can be used for characterizing the neighborhood of singularities as well as the robot performances (e.g. the condition number, the dexterity [53], etc.). Here, as generally the visual servoing is used for improving the robot accuracy, it is proposed to use accuracy as an index for the characterization of singularity proximity. Obviously, other criteria could be chosen, but the global approach for selecting the legs will remain the same.

For characterizing the performances of the robot in terms of accuracy, the following model can be used. From (7), and using the first order approximation of the forward geometric model [53], it is possible to write

$$ {}^{c}\delta\underline{\mathbf{p}} = \mathbf{M}^{T+}\, {}^{c}\delta\underline{\mathbf{u}} \tag{A.1}$$

where ${}^c\delta\underline{\mathbf{p}}$ is the platform positioning error (expressed in the camera frame), ${}^c\delta\underline{\mathbf{u}}$ is the error on the observation of the leg direction (expressed in the camera frame), and $\mathbf{M}^{T+}$ is the pseudo-inverse of the matrix $\mathbf{M}^T$ that can be obtained by stacking the matrices $\mathbf{M}_i^T$ of the observed legs. Obviously, when using the leg direction based visual servoing, this matrix is the Jacobian matrix of the hidden robot and, as a result, will degenerate near the singular configurations presented in Section 3.

The error on the platform can be expressed in the robot frame using:

$$ {}^b\delta\underline{\mathbf{p}} = {}^b\overline{\mathbf{R}}_c \, {}^c\delta\underline{\mathbf{p}} = {}^b\overline{\mathbf{R}}_c \, \mathbf{M}^{T+} \, {}^c\delta\underline{\mathbf{u}} \tag{A.2} $$

where

$$ {}^b\overline{\mathbf{R}}_c = \begin{bmatrix} {}^b\mathbf{R}_c & \mathbf{0} \\ \mathbf{0} & {}^b\mathbf{R}_c \end{bmatrix} \tag{A.3} $$

in which ${}^b\mathbf{R}_c$ is the 3×3 rotation matrix between the base and camera frames.

It should be mentioned here that it is decided to develop a simple model for computing the robot accuracy, but any other more realistic models can be used (e.g. non linearized models [52], models that take into account flexibilities [54], clearances [55], etc.). However, this model is enough for giving a global idea the accuracy problems of the controlled robots, as was done in Section 5.

Once the performance criteria are chosen, it is necessary to define a methodology that can help select the legs to observe. Such a methodology is developed below.

Appendix A.2. Proposition of a methodology for selecting the legs to observe

The Sections 4 and Appendix A.1 showed the importance of the legs chosen for the control scheme: the singularities of the hidden robots depend on the chosen observed legs and the accuracy will be poor near them. Several questions naturally arise here. The first one concerns the number of legs to observe. In terms of accuracy, it is obvious that observing more legs than the minimal requested number, i.e. adding measurement redundancy, will tend to improve the pose accuracy of the robot. However:

- increasing the number of legs to observe leads to an increase of the computational time and may be applied with difficulty when high sampling periods are required. Thus, a compromise must be found between the sampling period and the computational time for any given application.
- it will be shown in Section 5 that, even when all legs are observed, some singular configurations may still exist, with areas of poor robot accuracy around them.

The second question is about the selection of the legs to observe. For example, in the case of the Quattro (Section 4.2.2), with only two legs among four to observe, six different 2–Π–{2–$\underline{U}U$} robots can be defined. What is thus the best hidden robot model to use?

If the control law proposed in Section 2 is applied, it is first necessary to guarantee that, for the used set of legs:

- obviously, the legs must be observable during the whole robot displacement.
- the initial and final robot configurations must be included in the same aspect. If not, the controller will not be able to converge to the desired end-effector pose, even if the observed leg directions do. In this last case, the problem can be solved by applying special trajectories that cross Type 2 singularities [56].

One problem here is to check that two robot configurations belong to the same aspect. This problem is complex for most of robots, but can be solved using some advanced tools such as Interval Analysis (*IA*) [2] or Cylindrical Algebraic Decomposition (*CAD*) [57].

Then, if accuracy is needed, the leg selection must guarantee the best final accuracy. To achieve this goal, the following procedure can be used:

1. knowing the leg orientations at the initial and final robot configurations, compute the solutions of the *fkp* of the hidden robots,

2. find, using the advanced tools mentioned previously, the solutions of the *fkp* that belong to the same assembly mode; if, for one given hidden robot, initial and final platform configurations do not belong to the same aspect, discard it; if no hidden robot exists for which initial and final configurations belong to the same aspect, the displacement is not feasible, except if special trajectories are planned as mentioned previously,
3. for all remaining virtual hidden robots, knowing the observation errors $\delta\underline{\mathbf{u}}$, compute the positioning error using (A.2); retain the set of legs that guarantee the best accuracy;
4. test the controller (in simulation) with the retained set of legs; if there is no problem of convergence and that the legs are observable during the whole displacement, the problem is solved; if not, discard this set of leg and redo point 3; if no hidden robot exists for which initial and final configurations belong to the same aspect, the displacement is not feasible, except with special trajectories.

Obviously, this methodology can be extended to any number of observed legs or modified by weighting the interaction matrices to obtain better robot properties. One should also be aware that instead of giving the initial and final robot configurations to the controller, it is better to define a trajectory between these two points in order to avoid crossing singularities inadvertently. In such a case, it is possible to check (numerically using any closeness-to-singularity criteria [53] or algebraically with *CAD* [57]) that the robot does not cross a singularity on the trajectory.

Appendix B. Kinematics of the Quattro

Appendix B.1. Usual inverse kinematics of the Quattro

The following notations are used:

- point B_i (C_i, resp.) is at the middle of segment $B_{i1}B_{i2}$ ($C_{i1}C_{i2}$, resp.) (Fig. 12(a)),
- point P, the controlled point of the platform, is the barycenter of points C_i; its coordinates are denoted as $\mathbf{x}_c$ and its velocity as $\boldsymbol{\tau}_c$,
- the platform orientation is parameterized by the angle ϕ between the axis $\mathbf{x}$ of the robot base frame and the vector $\overrightarrow{D_1D_2}$,
- $\mathbf{A}_i$ ($\mathbf{B}_i$, $\mathbf{C}_i$, resp.) is the vector of coordinates of point A_i (B_i, C_i, resp.),
- q_i is the angular coordinate of the actuator i, and is defined as the angle between the axis $\mathbf{x}_i$ (the projection of vector $\overrightarrow{A_iB_i}$ in the horizontal plane ($\mathbf{Oxy}$)) and $\overrightarrow{A_iB_i}$ around $\mathbf{y}_i$ (Fig. 12(a)),
- l_1 is the length of the proximal link, and l_2 the length of one rod of the parallelogram,

It is to be noticed that the Adept Quattro has the following geometric characteristics:

- $l_1 = 0.380$m, $l_2 = 0.825$m,
- ${}^b\mathbf{A}_i = 0.275\,[\cos\theta_i \;\; \sin\theta_i \;\; 0]^T$ (in meters), where $\theta_i = \{-3\pi/4, -\pi/4, \pi/4, 3\pi/4\}$ (in radians)
- ${}^b\overrightarrow{D_1C_1} = [-0.066 \;\; -0.048 \;\; 0]^T$, ${}^b\overrightarrow{D_1C_2} = [0.066 \;\; -0.048 \;\; 0]^T$, ${}^b\overrightarrow{D_2C_3} = [0.066 \;\; 0.048 \;\; 0]^T$ and ${}^b\overrightarrow{D_2C_4} = [-0.066 \;\; 0.048 \;\; 0]^T$ (in meters),
- ${}^b\overrightarrow{E_1D_1} = \overrightarrow{E_4D_2} = [0.057 \;\; 0 \;\; 0]^T$, ${}^b\overrightarrow{E_2D_1} = \overrightarrow{E_3D_2} = [-0.057 \;\; 0 \;\; 0]^T$ (in meters),
- ${}^b\overrightarrow{PE_2} = 0.043\,[\sin(\phi + \pi/2) \;\; -\cos(\phi + \pi/2) \;\; 0]^T$, ${}^b\overrightarrow{PE_2} = 0.043\,[\sin(\phi + \pi/2) \;\; \cos(\phi + \pi/2) \;\; 0]^T$ (in meters).

Moreover, the superscript 'i' will be used before the vectors to indicate that the vector coordinates are expressed in the leg local frame ($\mathbf{O}\mathbf{x}_i\mathbf{y}_i\mathbf{z}_i$). If no superscript is used, the vector is expressed in the base frame.

The usual inverse kinematics of the Quattro can be computed using the following loop-closure equations (Fig. 12(b)):

$$ {}^{i}\mathbf{C}_i - {}^{i}\mathbf{B}_i = l_2 {}^{i}\underline{\mathbf{u}}_i \tag{B.1} $$

where

$$ {}^{i}\mathbf{B}_i = {}^{i}\mathbf{A}_i + l_1 \begin{bmatrix} \cos q_i & 0 & \sin q_i \end{bmatrix}^T = {}^{i}\mathbf{A}_i + l_1 {}^{i}\underline{\mathbf{v}}_i \tag{B.2} $$

and

$$ {}^{i}\mathbf{C}_i = {}^{i}\mathbf{x}_c + {}^{i}\overrightarrow{PC_i} \tag{B.3} $$

Squaring both sides of (B.1) and introducing (B.2) leads to

$$ (x_{A_iC_i} - l_1 \cos q_i)^2 + y_{A_iC_i}^2 + (z_{A_iC_i} - l_1 \sin q_i)^2 - l_2^2 = 0 \tag{B.4} $$

where ${}^{i}\mathbf{C}_i - {}^{i}\mathbf{A}_i = [x_{A_iC_i}, y_{A_iC_i}, z_{A_iC_i}]^T$. (B.4) can be finally solved as a second order polynomial in $\tan(q_i/2)$ by replacing $\cos q_i$ by $\left(1 - t_i^2\right) / \left(1 + t_i^2\right)$ and $\sin q_i$ by $2t_i / \left(1 + t_i^2\right)$, where $t_i = \tan(q_i/2)$. Skipping all mathematical derivations, it comes that:

$$ q_i = 2 \tan^{-1} \left(\frac{-\beta_i \pm \sqrt{\alpha_i^2 + \beta_i^2 - \gamma_i^2}}{\gamma_i - \alpha_i} \right) \tag{B.5} $$

where

$$ \alpha_i = -2 l_1 x_{A_iC_i}, \beta_i = -2 l_1 z_{A_iC_i} \tag{B.6} $$

$$ \gamma_i = x_{A_iC_i}^2 + y_{A_iC_i}^2 + z_{A_iC_i}^2 + l_1^2 - l_2^2 \tag{B.7} $$

The first-order kinematics that relates the platform translational velocity $\boldsymbol{\tau}_c$ to the actuator velocities can be obtained through the differentiation of (B.4) with respect to time and can be expressed as:

$$ \mathbf{A}\boldsymbol{\tau}_c + \mathbf{B}\dot{\mathbf{q}} = \mathbf{0} \tag{B.8} $$

where the i-th line of $\mathbf{A}$ can be written as

$$ \mathbf{a}_i = l_2 \underline{\mathbf{u}}_i^T \tag{B.9} $$

and $\mathbf{B}$ is a diagonal matrix whose i-th diagonal term is

$$ b_i = l_1 l_2 {}^{i}\underline{\mathbf{u}}_i^T {}^{i}\underline{\mathbf{v}}_i^{\perp}, \; {}^{i}\underline{\mathbf{v}}_i^{\perp} = \begin{bmatrix} -\sin q_i & 0 & \cos q_i \end{bmatrix}^T \tag{B.10} $$

It should be mentioned that $\mathbf{A}$ is a (4×3) rectangular matrix. As a result,

$$ \dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\boldsymbol{\tau}_c = \mathbf{J}_{inv}\boldsymbol{\tau}_c, \text{ or also } \boldsymbol{\tau}_c = \mathbf{J}_{inv}^{+}\dot{\mathbf{q}} \tag{B.11} $$

where $\mathbf{J}_{inv}^{+}$ is the pseudo-inverse of $\mathbf{J}_{inv}$.

Appendix B.2. Kinematics of the Quattro using leg observation

The servoing of the Adept Quattro robot using leg observation proposes to observe the parallelogram direction $\underline{\mathbf{u}}_i$ to control the robot displacements. $\underline{\mathbf{u}}_i$ can be obtained directly from (B.1)

$$\underline{\mathbf{u}}_i = (\mathbf{C}_i - \mathbf{B}_i)/l_2 \tag{B.12}$$

Introducing (B.2) into (B.12) and differentiating (B.12) with respect to time leads to:

$$\dot{\underline{\mathbf{u}}}_i = \left(\boldsymbol{\tau}_c - l_1 \underline{\mathbf{v}}_i^{\perp}\, \dot{q}_i\right)/l_2 \tag{B.13}$$

Finally, from (B.11), it comes that:

$$\dot{\underline{\mathbf{u}}}_i = \left(\mathbf{I}_3 + l_1 \underline{\mathbf{v}}_i^{\perp} \mathbf{a}_i / b_i\right)/l_2\, \boldsymbol{\tau}_c = \mathbf{M}_i^T\, \boldsymbol{\tau}_c \tag{B.14}$$

where $\mathbf{I}_3$ is the (3×3) identity matrix and matrix $\mathbf{M}_i^T$ is called the interaction matrix. These equations are valuable as long as $b_i \neq 0$ ($b_i = 0$ is a Type 1 singularity condition).

Note that the equation (B.14) requires the computation of the input joint variables q_i which can be estimated through the observation of the leg direction only (without any use of the encoder measurement).

It can be proven that the matrix $\mathbf{M}_i^T$ is of rank 2. As a result, a minimum of two independent legs is necessary to control the end-effector pose. An interaction matrix $\mathbf{M}^T$ can then be obtained by stacking the matrices $\mathbf{M}_i^T$ of k legs ($k = 2...4$). The conditions for the rank deficiency of the interaction matrix have been presented in Section 4.2.2.

The previous equations characterize the inverse kinematics of the hidden robot models of the Quattro.

It should be mentioned that the equations for the forward kinematics are not given here for reason of paper compactness as they are tedious. However, the *fkp* can be solved using the proposed geometric approach presented in Section 4.2.2.

Acknowledgements

This work was supported by the French ANR project ARROW (ANR-2011BS3-006-01), by the French ANR project EquipEx RobotEx (ANR-10-EQPX-44), by the European Union Program "Compétitivité Régionale et Emploi 2007–2013" (FEDER – Région Auvergne) and by the EU project Feder RobotEx.

The authors would like to thank Michel Coste from the Institut de Recherche Mathématique de Rennes (University of Rennes, France) for his smart advices about the use of the Bézout theorem and the definition of the Bohemian Dome.

References

[1] T. Leinonen, Terminology for the theory of machines and mechanisms, Mechanism and Machine Theory 26.

[2] J. Merlet, Parallel Robots, 2nd Edition, Springer, 2006.

[3] D. Dementhon, L. Davis, Model-based object pose in 25 lines of codes, International Journal of Computer Vision 15 (1995) 123–141.

[4] M. Dhome, M. Richetin, J. Lapresté, G. Rives, Determination of the attitude of 3-d objects from a single perspective view, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (12) (1989) 1265–1278.

[5] D. Lowe, Three-dimensional object recognition from single two-dimensional images, Artificial Intelligence 31 (1987) 355–394.

[6] B. Espiau, F. Chaumette, P. Rives, A new approach to visual servoing in robotics, IEEE Transactions on Robotics and Automation 8 (3).

[7] R. Horaud, F. Dornaika, B. Espiau, Visually guided object grasping, IEEE Transactions on Robotics and Automation 14 (4) (1998) 525–532.

[8] P. Martinet, J. Gallice, D. Khadraoui, Vision based control law using 3D visual features, in: Proceedings of the World Automation Congress, WAC96, Robotics and Manufacturing Systems, Vol. 3, Montpellier, France, 1996, pp. 497–502.

[9] A. Traslosheros, J. Sebastian, L. Angel, F. Roberti, R. Carelliz, Visual servoing of a parallel robot system, in: IEEE International Symposium on Intelligent Signal Processing, 2007.

[10] Z. Qi, J. McInroy, Improved image based visual servoing with parallel robot, Journal of Intelligent Robot Systems 53 (2008) 359–379.

[11] R. Garrido, A. Soria, M. Trujano, Visual pid control of a redundant parallel robot, in: 5th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2008), 2008.

[12] M. Trujano, R. Garrido, A. Soria, Robust visual control of parallel robots under uncertain camera orientation, International Journal of Advanced Robotic Systems 9.

[13] N. Andreff, A. Marchadier, P. Martinet, Vision-based control of a Gough-Stewart parallel mechanism using legs observation, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'05, Barcelona, Spain, 2005, pp. 2546–2551.

[14] V. Gough, S. Whitehall, Universal tyre test machine, in: Proceedings of the FISITA 9th International Technical Congress, 1962, pp. 117–317.

[15] N. Andreff, T. Dallej, P. Martinet, Image-based visual servoing of gough-stewart parallel manipulators using legs observation, International Journal of Robotics Research 26 (7) (2007) 677–687.

[16] E. Özgür, N. Andreff, P. Martinet, Dynamic control of the quattro robot by the leg edgels, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA11, Shanghai, China, 2011.

[17] N. Andreff, P. Martinet, Vision-based kinematic modelling of some parallel manipulators for control purposes, in: Proceedings of EuCoMeS, the First European Conference on Mechanism Science, Obergurgl, Austria, 2006.

[18] F. Chaumette, The Confluence of Vision and Control, no. 237 in LNCIS, Springer-Verlag, 1998, Ch. Potential problems of stability and convergence in image-based and position-based visual servoing, pp. 66–78.

[19] H. Michel, P. Rives, Singularities in the determination of the situation of a robot effector from the perspective view of 3 points, Tech. rep., INRIA (1993).

[20] S. Briot, P. Martinet, Minimal representation for the control of Gough-Stewart platforms via leg observation considering a hidden robot model, in: Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013), Karlsruhe, Germany, 2013.

[21] S. Caro, W. Khan, D. Pasini, J. Angeles, The rule-based conceptual design of the architecture of serial schonflies-motion generators, Mechanism and Machine Theory 45 (2) (2010) 251–260.

[22] N. Andreff, B. Espiau, R. Horaud, Visual servoing from lines, International Journal of Robotics Research 21 (8) (2002) 679–700.

[23] J. Plücker, On a new geometry of space, Philosophical Transactions of the Royal Society of London 155 (1865) 725–791.

[24] F. Chaumette, La commande des robots manipulateurs, Hermès, 2002.

[25] X. Kong, C. Gosselin, Type Synthesis of Parallel Mechanisms, Springer, 2007.

[26] G. Gogu, Structural Synthesis of Parallel Robots, Springer, 2008.

[27] J. Hervé, Lie group of rigid body displacements, a fundamental tool for mechanism design, Mechanism and Machine Theory 34 (5) (1999) 719–730.

[28] M. Carricato, V. Parenti-Castelli, Singularity-free fully-isotropic translational parallel manipulators, International Journal of Robotics Research 21 (2) (2002) 161–174.

[29] X. Kong, C. Gosselin, A class of 3-dof translational parallel manipulators with linear input-output equations, in: Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators, Québec City, QC, Canada, 2002, pp. 3–4.

[30] G. Gogu, Structural synthesis of fully-isotropic translational parallel robots via theory of linear transformations, European Journal of Mechanics. A/Solids 23 (6) (2004) 1021–1039.

[31] C. Gosselin, J. Angeles, Singularity analysis of closed-loop kinematic chains, IEEE Transactions on Robotics and Automation 6 (3) (1990) 281–290.

[32] D. Zlatanov, I. Bonev, C. Gosselin, Constraint singularities of parallel mechanisms, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002), 2002.

[33] I. Bonev, D. Zlatanov, C. Gosselin, Singularity analysis of 3-dof planar parallel mechanisms via screw theory, ASME Journal of Mechanical Design 125 (3) (2003) 573–581.

[34] P. Ben-Horin, M. Shoham, Singularity analysis of a class of parallel robots based on grassmanncayley algebra, Mechanism and Machine Theory 41 (8) (2006) 958–970.

[35] S. Caro, G. Moroz, T. Gayral, D. Chablat, C. Chen, Singularity analysis of a six-dof parallel manipulator using grassmann-cayley algebra and grobner bases, in: Proceedings of the Symposium on Brain, Body and Machine, Montreal, QC, Canada, 2010.

[36] S. Briot, I. Bonev, D. Chablat, P. Wenger, V. Arakelian, Self motions of general 3-rpr planar parallel robots, International Journal of Robotics Research 27 (7) (2008) 855–866.

[37] L. Tsai, Kinematics and optimization of a spatial 3-upu parallel manipulator, ASME Journal of Mechanical Design 122 (2000) 439–446.

[38] R. Clavel, Device for the movement and positioning of an element in space (December 1990).

[39] V. Nabat, M. de la O Rodriguez, O. Company, S. Krut, F. Pierrot, Par4: very high speed parallel robot for pick-and-place, in: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)., 2005.

[40] D. Chablat, P. Wenger, Architecture optimization of a 3-dof parallel mechanism for machining applications, the Orthoglide, IEEE Transactions on Robotics and Automation 19 (3) (2003) 403–410.

[41] I. Bonev, Direct kinematics of zero-torsion parallel mechanisms, in: Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008), 2008.

[42] A. Wolf, M. Shoham, F. Park, Investigation of singularities and self-motions of the 3-UPU robot, in: Advances in Robot Kinematics, Dordrecht, Germany, 2002.

[43] C. Tischler, K. Hunt, A. Samuel, A spatial extension of cardanic movement: its geometry and some derived mechanisms, Mechanism and Machine Theory 33 (1998) 1249–1276.

[44] S. Krut, O. Company, M. Benoit, H. Ota, F. Pierrot, I4: A new parallel mechanism for Scara motions, in: Proceedings of the 2003 International Conference on Robotics and Automation (ICRA 2013), 2003.

[45] L. Tsai, S. Joshi, Comparison study of architectures of four 3 degree-of-freedom translational parallel manipulators, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2001), 2001.

[46] F. Pierrot, M. Uchiyama, P. Dauchez, A. Fournier, A new design of a 6-dof parallel robot, Proceedings of the 23rd International Symposium on Industrial Robots. Jounrla of Robotics and Mechatronics (1990) 308–315.

[47] M. Honegger, A. Codourey, E. Burdet, Adaptive control of the Hexaglide, a 6 dof parallel manipulator, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1997), 1997.

[48] O. Company, F. Pierrot, Modelling and preliminary design issues of a 3-axis parallel machine-tool, Mechanisms and Machine Theory 37 (2002) 1325–1345.

[49] M. Tale Masouleh, C. Gosselin, M. Husty, D. Walter, Forward kinematic problem of 5-RPUR parallel mechanisms (3T2R) with identical limb structures, Mechanism and Machine Theory 46 (2011) 945–959.

[50] E. Bézout, Recherches sur le degré des équations résultantes de l'évanouissement des inconnues, Histoire de l'Académie Royale des Sciences, 1764.

[51] S. Briot, P. Martinet, V. Rosenzveig, The hidden robot: an efficient concept contributing to the analysis of the controllability of parallel robots in advanced visual servoing techniques, IEEE Transactions on Robotics 31 (6) (2015) 1337–1352.

[52] S. Briot, I. Bonev, Accuracy analysis of 3T1R fully-parallel robots, Mechanism and Machine Theory 45 (5) (2010) 695–706.

[53] J. Merlet, Jacobian, manipulability, condition number, and accuracy of parallel robots, ASME Transactions Journal of Mechanical Design 128 (1) (2006) 199–206.

[54] A. Pashkevich, D. Chablat, P. Wenger, Stiffness analysis of overconstrained parallel manipulators, Mechanism and Machine Theory 44 (5) (2009) 966–982.

[55] N. Binaud, P. Cardou, S. Caro, P. Wenger, The kinematic sensitivity of robotic manipulators to joint clearances, in: Proceedings of ASME Design Engineering Technical Conferences, Montreal, QC, Canada, 2010.

[56] S. Briot, V. Arakelian, Optimal force generation of parallel manipulators for passing through the singular positions, International Journal of Robotics Research 27 (8) (2008) 967–983.

[57] D. Chablat, G. Moroz, P. Wenger, Uniqueness domains and non singular assembly mode changing trajectories, in: Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA 2011), Shanghai, China, 2011.