



HAL
open science

Minimal Representation for the Control of the Adept Quattro with Rigid Platform via Leg Observation Considering a Hidden Robot Model

Victor Rosenzveig, Sébastien Briot, Philippe Martinet

► **To cite this version:**

Victor Rosenzveig, Sébastien Briot, Philippe Martinet. Minimal Representation for the Control of the Adept Quattro with Rigid Platform via Leg Observation Considering a Hidden Robot Model. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), Nov 2013, Tokyo Big Sight, Japan. hal-00839839

HAL Id: hal-00839839

<https://hal.science/hal-00839839>

Submitted on 25 Jun 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Minimal Representation for the Control of the Adept Quattro with Rigid Platform via Leg Observation Considering a Hidden Robot Model

Victor Rosenzweig¹, Sébastien Briot² and Philippe Martinet¹

Abstract—Previous works on the Gough-Stewart (GS) platform have shown that its visual servoing using the observation of its leg directions was possible by observing only three of its six legs but that the convergence to the desired pose was not guaranteed. This can be explained by considering that the visual servoing of the leg direction of the GS platform was equivalent to controlling another robot, the 3-UPS that has assembly modes and singular configurations different from those of the GS platform. Considering this hidden robot model allowed the simplification of the singularity analysis of the mapping between the leg direction space and the Cartesian space.

In this paper, the work on the definition of the hidden robot models involved in the visual servoing using the observation of the robot leg directions is extended to another robot, the Adept Quattro. It will be shown that the hidden robot model is completely different from the model involved in the control of the GS platform. Therefore, the results obtained for the GS platform are not valuable for this robot. The hidden robot has assembly modes and singular configurations different from those of the Quattro. An accuracy analysis is performed to show the importance of the leg selection. All these results are validated on a Quattro simulator created using ADAMS/Controls and interfaced with Matlab/Simulink.

I. INTRODUCTION

Compared to serial robots, parallel robots are stiffer and can reach higher speeds and accelerations [1]. However, their control is more complex because of the highly coupled mechanical structure 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 (see [2] for a long list of references). 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 proved that the robot end-effector pose can be effectively estimated by vision. The most

This work was supported by the French ANR ARROW (ANR 2011BS3 006 01). The authors would also like to thank Nicolas Bouton from the IFMA (Institut Pascal) of Clermont-Ferrand for his great help with the numerical mockup of the Quattro.

¹V. Rosenzweig and P. Martinet are with the LUNAM University, Ecole Centrale de Nantes and the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), 44321 Nantes France (Victor.Rosenzweig@ircryn.ec-nantes.fr, Philippe.Martinet@ircryn.ec-nantes.fr)

²S. Briot is with the French CNRS and IRCCyN, 44321 Nantes France (Sebastien.Briot@ircryn.ec-nantes.fr)

common approach consists of the direct observation of the end-effector pose [3], [4], [5]. However, some applications prevent the observation of the end-effector of a parallel mechanism by vision. 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 [6] where vision was used to derive a visual servoing scheme based on the observation of a Gough-Stewart (GS) parallel robot [7]. In that method, the leg directions were chosen as visual primitives and control was derived based on their reconstruction from the image. In this work, it has also been observed that:

- 1) the GS platform can be controlled using the observation of only three leg directions (arbitrarily chosen among its six legs), and that
- 2) in some cases, the GS platform does not converge to the desired end-effector pose (even if the observed leg directions did)

without finding some concrete explanations to these points.

Recently, two of the authors of the present paper have demonstrated in [8] that these two points could be explained by considering that the visual servoing of the leg direction of the GS platform was equivalent to controlling another robot, the 3-UPS (where U stands for an actuated universal joint, P and S for passive prismatic and spherical joints, respectively) that has assembly modes and singular configurations different from those of the GS platform. Considering this hidden robot model allowed the simplification of 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 and it was possible to find simple geometric conditions for the singular configurations.

In this paper, the visual servoing of the Adept Quattro using leg observation is considered. It will be shown that another hidden robot model (with a kinematic chain completely different from the 3-UPS, i.e. the hidden model for the vision based control of the GS platform) is involved in this kind of control. Consequently, the assembly modes and singularity conditions of this hidden robot are totally different from those of the 3-UPS and the results obtained in [8] are not valuable for the Quattro.

II. CONTROL OF THE ADEPT QUATTRO WITH RIGID PLATFORM USING LEG OBSERVATION

This part aims at presenting the servoing of the Adept Quattro robot with rigid platform using leg observation.

A. Kinematics of the Quattro with rigid platform

The Quattro (Fig. 1) is composed of four identical kinematic chains (legs), that carry the articulated moving platform. Each of the 4 kinematic chains is actuated from the base by a rotary motor, located at point A_i . This motor rotates a proximal link that is linked to an articulated parallelogram, itself attached to the moving platform. The parallelogram consists of two slim and cylindrical shaped rods fitted with ball-joints at points B_{i1} , B_{i2} , C_{i1} and C_{i2} . As a result, each kinematic chain is a $R(2-SS)$ leg (where R stands for an actuated revolute joint).

There exist two versions of the Quattro: one with a rigid (non-articulated) moving platform, thus leading to a redundant parallel robot with 3 translational degrees of freedom (*dof*) of the end-effector, and the other with an articulated platform that allows a supplementary rotation of the end-effector around the vertical axis z . This paper focuses only on the Quattro with the rigid platform, i.e. with 3 translational *dof*.

In the following of the paper, the notations detailed below are used:

- point B_i (C_i , resp.) is at the middle of segment $B_{i1}B_{i2}$ ($C_{i1}C_{i2}$, resp.) (Fig 1(b)),
- point P , the controlled point of the platform, is the barycentre of points C_i ; its coordinates are denoted as \mathbf{P} and its velocity as τ_p ,
- \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 x_i (the projection of vector A_iB_i in the horizontal plane (Ox_iy_i)) and A_iB_i around y_i (Fig. 1(b)),
- l_1 is the length of the proximal link, and l_2 the length of one rod of the parallelogram,

Moreover, the superscript ' i ' will be used before the vectors to indicate that the vector coordinates are expressed in the leg local frame ($Ox_iy_iz_i$). 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. 1):

$${}^i\mathbf{C}_i - {}^i\mathbf{B}_i = l_2 {}^i\mathbf{u}_i \quad (1)$$

where

$${}^i\mathbf{B}_i = {}^i\mathbf{A}_i + l_1 [\cos q_i \quad 0 \quad \sin q_i]^T = {}^i\mathbf{A}_i + l_1 {}^i\mathbf{v}_i \quad (2)$$

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

$$(x_{A_iC_i} - l_1 \cos q_i)^2 + y_{A_iC_i}^2 + (z_{A_iC_i} - l_1 \sin q_i)^2 - l_2^2 = 0 \quad (3)$$

where ${}^i\mathbf{C}_i - {}^i\mathbf{A}_i = [x_{A_iC_i}, y_{A_iC_i}, z_{A_iC_i}]^T$. (3) can be finally solved as a second order polynomial in $\tan(q_i/2)$ by replacing $\cos q_i$ by $(1 - t_i^2) / (1 + t_i^2)$ and $\sin q_i$ by $2t_i / (1 + t_i^2)$, where $t_i = \tan(q_i/2)$. Skipping all mathematical derivations, it comes that:

$$q_i = 2 \tan^{-1} \left(\frac{-\beta_i \pm \sqrt{\alpha_i^2 + \beta_i^2 - \gamma_i^2}}{\gamma_i - \alpha_i} \right) \quad (4)$$

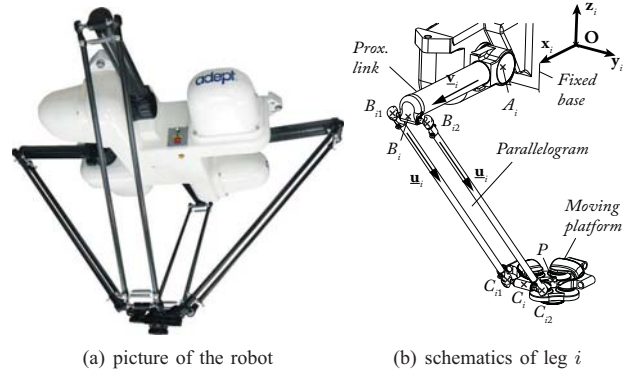


Fig. 1. The Adept Quattro (courtesy of Adept).

where

$$\begin{aligned} \alpha_i &= -2l_1 x_{A_iC_i}, \beta_i = -2l_1 z_{A_iC_i} \\ \gamma_i &= x_{A_iC_i}^2 + y_{A_iC_i}^2 + z_{A_iC_i}^2 + l_1^2 - l_2^2 \end{aligned} \quad (5)$$

The first-order kinematics that relates the platform translational velocity τ_p to the actuator velocities can be obtained through the differentiation of (3) with respect to time and can be expressed as:

$$\mathbf{A}\tau_p + \mathbf{B}\dot{\mathbf{q}} = \mathbf{0} \quad (6)$$

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

$$\mathbf{a}_i = l_2 \mathbf{u}_i^T \quad (7)$$

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

$$b_i = l_1 l_2 {}^i\mathbf{u}_i^T {}^i\mathbf{v}_i^\perp, \quad {}^i\mathbf{v}_i^\perp = [-\sin q_i \quad 0 \quad \cos q_i]^T \quad (8)$$

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

$$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}\tau_p = \mathbf{J}_{inv}\tau_p, \quad \text{or also } \tau_p = \mathbf{J}_{inv}^+\dot{\mathbf{q}} \quad (9)$$

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

B. Kinematics of the the Quattro using leg observation

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

$$\mathbf{u}_i = (\mathbf{C}_i - \mathbf{B}_i) / l_2 \quad (10)$$

Introducing (2) into (10) and differentiating (10) with respect to time leads to:

$$\dot{\mathbf{u}}_i = (\tau_p - l_1 \mathbf{v}_i^\perp \dot{q}_i) / l_2 \quad (11)$$

Finally, from (9), it comes that:

$$\dot{\mathbf{u}}_i = (\mathbf{I}_3 + l_1 \mathbf{v}_i^\perp \mathbf{a}_i / b_i) / l_2 \tau_p = \mathbf{M}_i^T \tau_p \quad (12)$$

where \mathbf{I}_3 is the 3×3 identity matrix and matrix \mathbf{M}_i^T is called the interaction matrix.

It can be proved that matrix \mathbf{M}_i^T is of rank 2 [6]. 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 ($k = 2...4$).

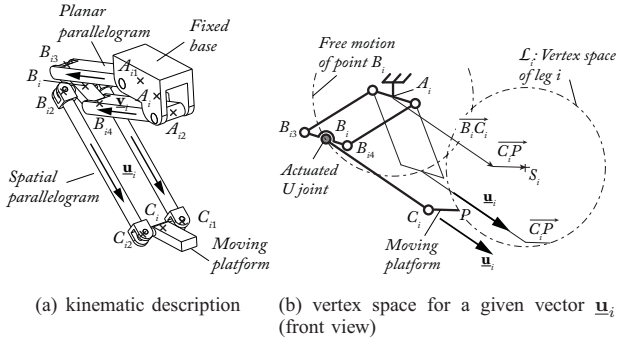


Fig. 2. A $\Pi(2-\underline{U})$ leg.

C. Control scheme and interaction matrix

Visual servoing is based on the so-called interaction matrix \mathbf{M}^T [9] which relates the instantaneous relative motion $T_c = {}^c\tau_c - {}^c\tau_s$ between the camera and the scene, to the time derivative of the vector s of all the visual primitives that are used through:

$$\dot{s} = \mathbf{M}_{(s)}^T T_c \quad (13)$$

where ${}^c\tau_c$ and ${}^c\tau_s$ are respectively the twists of the camera and the scene, both expressed in \mathcal{R}_c , i.e. the camera frame.

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

$$T_c = \lambda \hat{\mathbf{M}}_{(s)}^{T+} e(s, s_d) \quad (14)$$

where T_c is used as a pseudo-control variable and the upperscript $+$ 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:

$$\mathbf{e}_i = \underline{\mathbf{u}}_i \times \underline{\mathbf{u}}_{di} \quad (15)$$

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

Finally, a control is chosen such that E , the vector stacking the errors \mathbf{e}_i associated to k legs ($k = 2 \dots 4$), decreases exponentially, i.e. such that

$$\dot{E} = -\lambda E \quad (16)$$

Then, introducing $\mathbf{N}_i^T = -[\underline{\mathbf{u}}_{di}]_{\times} \mathbf{M}_i^T$ (where $[\dots]_{\times}$ is the antisymmetric matrix associated to a 3D vector [5]), the combination of (12), (15) and (16) gives

$$\tau_p = -\lambda \mathbf{N}^{T+} E \quad (17)$$

where \mathbf{N}^T can be obtained by stacking the matrices \mathbf{N}_i^T of k legs ($k = 2 \dots 4$).

This expression can be transformed into the control joint velocities using (9):

$$\dot{\mathbf{q}} = -\lambda \mathbf{J}_{inv} \mathbf{N}^{T+} E \quad (18)$$

In the next section, it is shown that observing the displacement of the leg directions $\underline{\mathbf{u}}_i$ is intrinsically equivalent to controlling another robot, different from the Quattro.

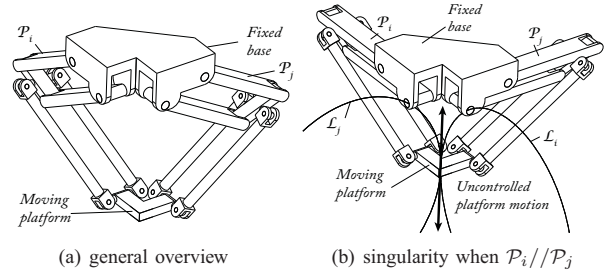


Fig. 3. A $2-\Pi(2-\underline{U})$ robot.

III. DESCRIPTION OF THE HIDDEN ROBOT MODEL

A. Description of the hidden robot architecture

In [8], it has been demonstrated that the visual servoing of the leg directions of the GS platform was equivalent to controlling another virtual robot composed of \underline{UPS} legs, for which the measure of the vectors $\underline{\mathbf{u}}_i$ was an image of the displacements of the actuated U joints. So, in a reciprocal manner, one could wonder what kind of virtual equivalent leg is involved in the visual servoing of the Quattro using leg directions.

The virtual equivalent leg for the Quattro must:

- guaranty that the kinematic relations (4), (9), (10) and (12) are still valid,
- ensure that, when only two legs are observed, the equivalent robot made of the two virtual legs is able to keep the platform orientation constant,
- contain active U joints that can be related to the measure of $\underline{\mathbf{u}}_i$, i.e. *these joints should move if and only if the value of $\underline{\mathbf{u}}_i$ is changing.*

Considering these three points, it could be proved by a simple kinematic analysis that the equivalent leg has a $\Pi(2-\underline{U})$ architecture, (where Π stands for a planar parallelogram linkage joint – Fig. 2(a)). The Π joint keeps constant the orientation of the rod $B_{i3}B_{i4}$ (Fig. 2(b)) and guaranties that, as long as the vector $\underline{\mathbf{u}}_i$ is set at a constant value, the first U joints (that link the planar and spatial parallelograms) are not moving. The passive U joints linked to the platform prevent its rotation when only two legs are used. Thus, observing the direction of the Quattro leg remains not to control the displacement of a $R(2-SS)$ leg but of a virtual $\Pi(2-\underline{U})$ leg with the same geometric properties as the real leg ($\|\overrightarrow{A_i B_i}\| = l_1$, $\|\overrightarrow{B_i C_i}\| = l_2$ in Fig. 2).

It can be demonstrated that a $2-\Pi(2-\underline{U})$ robot (Fig. 3(a)) is fully-actuated. Therefore, this is the reason why it is possible to control the Quattro by observing the displacements of two of its four legs. This is equivalent to actuate a virtual $2-\Pi(2-\underline{U})$ robot with the same geometric properties as the Quattro (same attachment points, leg length, etc.), but with assembly modes and singular configurations that differ from those of the Quattro. They should be studied in order to avoid control problems.

B. Forward kinematic analysis

Using the developed form of (1) and (2), it comes that:

$$\begin{bmatrix} x_{A_i C_i} - l_2 u_i^x \\ y_{A_i C_i} - l_2 u_i^y \\ z_{A_i C_i} - l_2 u_i^z \end{bmatrix} = l_1 \begin{bmatrix} \cos q_i \\ 0 \\ \sin q_i \end{bmatrix} \quad (19)$$

where $\underline{\mathbf{u}}_i^T = [u_i^x \ u_i^y \ u_i^z]$. Rearranging the terms of (19) and developing the expressions, a set of equations can be obtained (for $i = 1 \dots 4$)

$$\begin{aligned} l_1^2 &= ({}^i x - x_{S_i})^2 + ({}^i z - z_{S_i})^2 \\ 0 &= {}^i y - y_{S_i} \end{aligned} \quad (20)$$

where ${}^i \mathbf{S}_i = [x_{S_i} \ y_{S_i} \ z_{S_i}]^T = {}^i \mathbf{A}_i + l_2 {}^i \underline{\mathbf{u}}_i + {}^i \overrightarrow{C_i P}$ and ${}^i \mathbf{P} = [{}^i x \ {}^i y \ {}^i z]^T$. Equations (20) are equations of circles, denoted as \mathcal{L}_i , of radius l_1 , located in planes $y = y_{S_i}$ and centred in point S_i of coordinates ${}^i \mathbf{S}_i$. These circles represent the vertex space of the tip of the leg i , when the vector $\underline{\mathbf{u}}_i$ is fixed and the planar parallelogram passively moving (Fig. 2(b)) [1]. As a result, the forward kinematic problem (*fkp*) is equivalent to finding the intersection between the circles \mathcal{L}_i of the observed legs. In conclusion, it may exist 0, 1 unique solution, 1 double solution (singularity condition), 2 distinct solutions or infinite number of solutions (if the circles are superposed, which is also a singularity condition) to the *fkp*.

Thus, the 2- Π (2- \underline{UU}) robot can have up to two distinct assembly modes that are different from those of the Quattro. The existence of these assembly modes explains the second point presented in the introduction, i.e. the non systematic convergence of the end-effector of the observed robots to the desired pose, even if the observed leg directions do. A numerical example of this phenomenon will be presented in the section V.

C. Singularity analysis

Three types of singular configurations, related to the degeneracy of the first-order kinematics, may appear for robots [10]:

- 1) the Type 1 singularities where the robot loses at least one degree of freedom;
- 2) the Type 2 singularities where there is the apparition of some uncontrollable motions. In their neighborhood, the platform accuracy considerably decreases;
- 3) the Type 3 singularities, where both Type 1 and Type 2 singularities encounter.

There exist other types of singularities, such as the constraint singularities [11], but they are due to passive constraint degeneracy only, and are not involved in the mapping between the leg directions and the Quattro end-effector space.

For the 2- Π (2- \underline{UU}) robot, Type 1 singularities appear when $\overrightarrow{A_i B_i}$ and $\overrightarrow{B_i C_i}$ are colinear. In such cases, the robot reaches its workspace boundary. 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 circles \mathcal{L}_i and \mathcal{L}_j have a common tangent at their intersection point

and the robot gains one uncontrollable *dof* along this tangent (Fig. 3(b)).

Obviously, the singularity loci vary depending on the leg chosen for the Quattro control. Therefore, it is extremely important, for having the best performances of the controller, to make an optimal selection of the legs to observe. This is the topic of the next section.

Finally, it should be mentioned that the singularities of the 2- Π (2- \underline{UU}) robot are not physical singularities, in the sense that they do not lead to uncontrollable free motions of the platform. However, they are representation singularities due to the mapping from the Cartesian space to the leg direction space [12].

IV. SELECTION OF THE CONTROLLED LEGS

Several indices can be used for characterizing the neighborhood of singularities (e.g. the condition number, the dexterity [13], etc.). Here, as generally, the visual servoing is used for improving the robot accuracy, it is proposed to use it as an index for the characterization of singularity proximity.

A. Accuracy analysis

From (12), and using the first order approximation of the forward geometric model [13], it is possible to write

$$\delta \underline{\mathbf{p}} = \mathbf{M}^{T+} \delta \underline{\mathbf{u}} \quad (21)$$

where $\delta \underline{\mathbf{p}}$ is the platform positioning error, $\delta \underline{\mathbf{u}}$ is the error on the observation of the leg direction, 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, this matrix is the Jacobian matrix of the equivalent 2- Π (2- \underline{UU}) robot and, as a result, will degenerate near the singularity configurations presented in section III-C. It should be mentioned here that it is decided to use a simple model for computing the robot accuracy, but any other more complicated models can be used (e.g. models that take into account flexibilities [14], clearances [15], etc.). However, this simple model is enough for our demonstration.

The Adept Quattro has the following characteristics:

- $l_1 = 0.380\text{m}$, $l_2 = 0.825\text{m}$,
- $\underline{\mathbf{A}}_i = 0.296 [\cos \theta_i \ \sin \theta_i \ 0]^T$ (in meters)
- $\overrightarrow{P C_i} = 0.102 [\cos \theta_i \ \sin \theta_i \ 0]^T$ (in meters)
- $\theta_i = \{-3\pi/4, -\pi/4, \pi/4, 3\pi/4\}$ (in radians)

For this mechanism, and for an error $\delta \underline{\mathbf{u}}_i$ defined such that the vector $\underline{\mathbf{u}}_i$ is contained in a cone of axis $\underline{\mathbf{u}}_{i0}$ and of half angle ϕ_i ($\underline{\mathbf{u}}_{i0}$ is the nominal value of $\underline{\mathbf{u}}_i$ and, in what follows, ϕ_i is taken equal to 0.01 deg. for each leg direction), let us compute the maximal positioning 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 $\{1, 2\}$ and $\{1, 3\}$ are observed) is plotted at Fig. 4.

On Fig. 4(a), it is possible to note that the maximal error varies very quickly, especially near singularity. On Fig. 4(b), things are different. The variation of the accuracy is very smooth. Thus, it can be concluded that the selection of the legs to observe is crucial for the final pose accuracy.

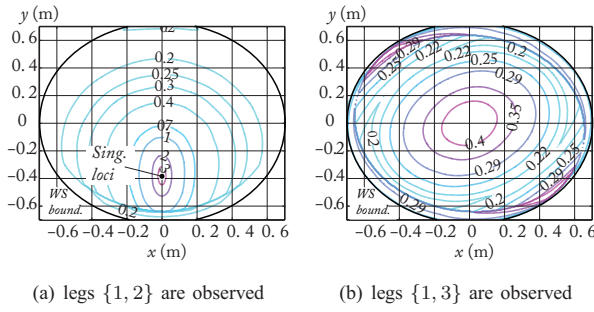


Fig. 4. Maximal pose error (in mm) for $z = -0.85$ m.

B. Discussion

The previous section showed the importance of the legs chosen for the control scheme. Several questions naturally arise here. The first one concerns the number of legs to observe. In terms of accuracy, it is obvious that observing three or four legs, i.e. adding measurement redundancy, will 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 low sampling periods are required. Thus, a compromise must be found between the sampling period and the computational time for any given application.

The second question is about the selection of the legs to observe. With only two legs among four to observe, as mentioned above, six different $2-\Pi(2-\underline{UU})$ robots can be defined. What is thus the best virtual robot model to use?

If the control law proposed in section II-C is applied, it is first necessary to guaranty 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 assembly mode of the virtual $2-\Pi(2-\underline{UU})$ robot. 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 [16] or encircle a cusp point [17].

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

- 1) knowing the four leg orientations at the initial and final Quattro configurations, compute the solutions of the f_{kp} of the six $2-\Pi(2-\underline{UU})$ robots,
- 2) find, using a procedure similar to the one proposed in [18] for all virtual $2-\Pi(2-\underline{UU})$ robots, the solutions of the f_{kp} that belong to the same assembly mode; if, for one given virtual robot, initial and final platform configurations do not belong to the same assembly mode, discard it; if it does not exist any $2-\Pi(2-\underline{UU})$ robot for which initial and final configurations belong to the same assembly mode, the displacement is not feasible, except if special trajectories are planned as mentioned previously,

- 3) for all remaining virtual $2-\Pi(2-\underline{UU})$ robots, knowing the observation error $\delta \underline{u}$, compute the positioning error using (21); retain the set of legs that guaranty 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 it does not exist any $2-\Pi(2-\underline{UU})$ robot for which initial and final configurations belong to the same assembly mode, the displacement is not feasible, except with special trajectories.

Obviously, this methodology can be extended when three or four legs are observed. One should also be aware that instead of given 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.

Finally, it is considered in this paper that the sensor measurement space is the same as the leg direction space. However, for example using a camera, the leg directions are not directly measured but rebuilt from the observation of the legs limbs projection in the 2D camera space [6]. Thus, for the leg reconstruction, the mapping between the camera space and the real 3D space is involved, and it is not free of singularities (see [19] for an example of mapping singularities). In the neighborhood of mapping singularities, the robot accuracy will also tend to decrease. As a result, this mapping should be considered in the accuracy computation and in the selection of the legs to observe.

V. SIMULATION RESULTS

In this section, simulations are performed on an ADAMS mockup of the Adept Quattro. This virtual mockup is connected to Matlab/Simulink via the module ADAMS/Controls. The controller presented in section II-C is applied (with $\lambda = 5$).

In the first simulations, the initial platform pose is equal to $\{x = 0.1m, y = 0.04m, z = -0.93m\}$ and the final platform pose is set to $\{x = 0m, y = -0.25m, z = -0.7m\}$. For going from the initial point to the final ones, two sets of observed leg directions are tested: $\{1, 2\}$ and $\{1, 3\}$. The results for the convergence of the leg directions are presented in Fig. 5. It can be shown that when the legs $\{1, 3\}$ 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 $\{x = 0m, y = -0.25m, z = -0.958m\}$ (Fig 6). Solving the forward geometric problem using (20) at the final desired robot configuration for legs $\{1, 2\}$, it can be demonstrated that two real assembly modes exist that are $\{x = 0m, y = -0.25m, z = -0.7m\}$ and $\{x = 0m, y = -0.25m, z = -0.958m\}$. This validates the theory presented in section III.

In the second simulation, the final point is changed to $\{x = 0m, y = -0.3m, z = -0.85m\}$ and a random noise of 0.01 deg. is added on the simulated measure of the leg directions. To show the importance of the leg selection on the

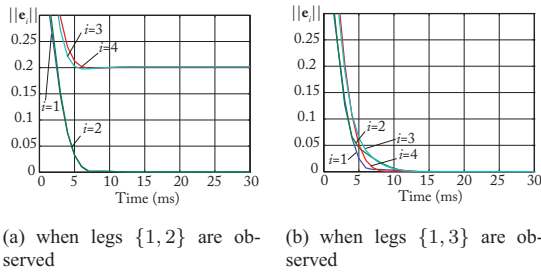


Fig. 5. Error norm on each leg $\|e_i\|$.

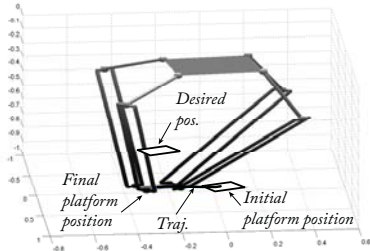


Fig. 6. Trajectory in space with initial, desired and final platform positions.

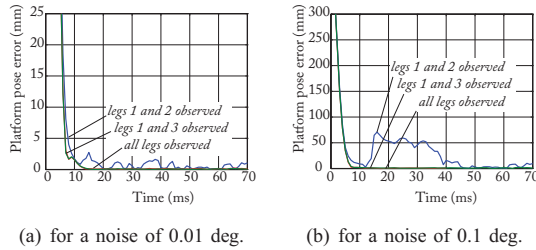


Fig. 7. Simulated platform pose error (in mm).

robot accuracy, it is decided to control the robot displacement using three different sets of legs: (i) legs $\{1, 2\}$, (ii) legs $\{1, 3\}$ and (iii) all legs. The results (Fig. 7(a)) show that, as presented in Fig. 4, the final platform pose accuracy is better when legs $\{1, 3\}$ are observed (around 0.22mm) than with legs $\{1, 2\}$ (around 1.6mm). When all legs are observed, the final pose error is a bit lower than when only two legs are observed (0.15mm). This seems normal. However, it can be proved that observing the four legs does not lead to a mapping without any singularities. A singular configuration still exists when all the planes \mathcal{P}_i for the four legs are horizontal, which can be difficult to detect without studying the equivalent robot model. Finally, for testing the robustness of the controller, the noise is multiplied by 10, but we can still observe a convergence of the platform to the desired pose (Fig. 7(b)).

VI. CONCLUSIONS

This paper presented new insights about the sensor-based control of the Adept Quattro. It has been shown that observing the leg directions of the Quattro involves controlling the displacement of a hidden robot which is based on $\Pi(2-\underline{UU})$

legs instead of $R(2-\underline{SS})$ legs. If only two legs are observed, the new equivalent robot becomes a $2-\Pi(2-\underline{UU})$ robot that has up to two distinct assembly modes that are different from those of the Quattro. Its conditions of singularities have been presented. It has been shown that the legs to observe should be chosen carefully in order to avoid inaccuracy problems.

All the presented results have been validated on a Quattro simulator created using ADAMS/Controls and interfaced with Matlab/Simulink. Future works will concern the generalization of the concept of hidden robot model for different classes of parallel robots and the experimental validations.

REFERENCES

- [1] J. Merlet, *Parallel Robots*. Springer, 2nd ed., 2006.
- [2] J. Merlet 2012.
- [3] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, 1992.
- [4] R. Horaud, F. Dornaika, and B. Espiau, "Visually guided object grasping," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 525–532, 1998.
- [5] P. Martinet, J. Gallice, and 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), pp. 497–502, May 1996.
- [6] N. Andreff, A. Marchadier, and 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), pp. 2546–2551, April 18–22 2005.
- [7] V. Gough and S. Whitehall, "Universal tyre test machine," in *Proceedings of the FISITA 9th International Technical Congress*, pp. 117–317, May 1962.
- [8] S. Briot and 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), May, 6–10 2013.
- [9] F. Chaumette, *La commande des robots manipulateurs*. Hermès, 2002.
- [10] C. Gosselin and J. Angeles, "Singularity analysis of closed-loop kinematic chains," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 3, pp. 281–290, 1990.
- [11] D. Zlatanov, I. Bonev, and C. Gosselin, "Constraint singularities of parallel mechanisms," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2002)*, May 2002.
- [12] O. Ma and J. Angeles, "Architecture singularities of parallel manipulators," *The International Journal of Robotics and Automation*, vol. 7, no. 1, pp. 23–29, 1992.
- [13] J. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *ASME Transactions Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, 2006.
- [14] A. Pashkevich, D. Chablat, and P. Wenger, "Stiffness analysis of over-constrained parallel manipulators," *Mechanism and Machine Theory*, vol. 44, no. 5, pp. 966–982, 2009.
- [15] N. Binaud, P. Cardou, S. Caro, and P. Wenger, "The kinematic sensitivity of robotic manipulators to joint clearances," in *Proceedings of ASME Design Engineering Technical Conferences*, (Montreal, QC, Canada), August 15–18 2010.
- [16] S. Briot and V. Arakelian, "Optimal force generation of parallel manipulators for passing through the singular positions," *International Journal of Robotics Research*, vol. 27, no. 8, pp. 967–983, 2008.
- [17] M. Zein, P. Wenger, and D. Chablat, "Non-singular assembly-mode changing motions for 3-RPR parallel manipulators," *Mechanism and Machine Theory*, vol. 43, no. 4, pp. 480–490, 2008.
- [18] I. Bonev, D. Chablat, and P. Wenger, "Working and assembly modes of the Agile Eye," in *Proceedings of the IEEE International Conference On Robotics And Automation (ICRA 1996)*, (Orlando, Florida, USA), 2006.
- [19] H. Michel and P. Rives, "Singularities in the determination of the situation of a robot effector from the perspective view of 3 points," tech. rep., INRIA, 1993.