Vector-Based Dynamic Modeling and Control of the Quattro Parallel Robot by means of Leg Orientations

Erol Ozgur, Nicolas Andreff, Philippe Martinet

***Abstract*— One of the key steps in high-speed control of a parallel robot is to define an efficient dynamic model. It is usually not easy to have such a model for parallel robots, since many of them have complex structures. Here, we propose a novel vector-based approach, which employs the leg orientations of the robot observed by a calibrated camera, to obtain a simplified inverse projective dynamic model. This vector-based methodology is a pioneer in the sense of solving the entire hard modeling problem through the knowledge of only a nominal set of image features: the edges and the edge velocities of the legs of the parallel robot. We verified our method on a simulator of the Quattro robot with a computed torque control.**

I. INTRODUCTION

Parallel robots have superior skills: they can reach high-speeds, show high-dynamic performances and achieve good repeatability [1]. These skills make them more attractive for many applications in comparison with serial robots.

However, their control is troublesome because of the complex mechanical structure, highly coupled joint motions due to the closed-loop kinematic chains and many factors such as clearances in passive joints, assembly errors, etc., which degrade stability and accuracy. Hence, to profit fully from these parallel mechanisms, one requires an efficient dynamic model, which should be purified from the complexity of the system, to use in the well-known *computed torque control* (CTC) [2]. Generally, in the literature, these models are considered to be written as a function of the joint coordinates due to the existence of only the actuator encoders as sensors for the measurement [3]. This makes derivation of simple models difficult without making assumptions [4] and overlooking some modeling errors in the mechanism.

What if we had additional sensors? The first attempt at this solution is made in [5] by introducing extra sensors, or so-called metrological redundancy, to simplify the kinematic models for easier control. So, having the inspiration of metrological redundancy, the immediate questions, which have to be answered to turn the tables on our side in the scenario, are "What actually should be sensed on the mechanism?" and "How can the modeling be adapted for the sensed data?" to have lighter models that will yield better control.

In this work, observation of the orientations of the slim and cylindrical shaped legs of the Quattro parallel robot is proposed as a solution. Indeed, it seems to be a good choice, since they play a crucial role in kinematics of a parallel robot [6]. Moreover, this observation will let us take advantage of a vector-based formulation rather than a formulation based on coordinates in dynamics as well.

In this paper, the previous work done in [7], for the kinematic modeling of the parallel robot (Section III), is pushed one step further by introducing a vector-based dynamic modeling based on leg orientations (Section IV). The introduced vector-based dynamic model: (i) has a more compact, readable and understandable written expression, (ii) in particular, suppresses the use of direct and inverse of the *sine* and *cosine* functions, (iii) is easier to implement, and (iv) has a more geometric flavour and hence lessened calculus. The leg orientations of the robot can be sensed with either a gyroscope, some special joint-sensors at extremities of the legs or by vision. Here, vision is chosen, since it is contactless, is easy to integrate and reduces the system calibration process by allowing all the measurements to be performed in a single reference frame. In Section V, it is shown that vision is rich enough to furnish the required variable sets of kinematics and dynamics for control. Finally, in Section VI, the proposed vector-based dynamic model, which makes use of vision directly in the internal control-loop to compensate for the dynamics, is validated on a simulator of the Quattro parallel robot. In order to make the terminology clear and ease understanding of the paper, we devote the next section to the geometry of the Quattro parallel robot and the notation used throughout the paper.

II. GEOMETRY OF THE MECHANISM

The Quattro is composed of four identical kinematic chains (legs), that carry the articulated travelling plate (nacelle). Each of the 4 kinematic chains is actuated from the base by a revolving motor, located at $\mathbf{P}_i$, and has two consecutive bodies (an arm and a forearm) linked with each other at $\mathbf{A}_i$. Each forearm consists of two slim and cylindrical shaped rods fitted with ball-joints (($\mathbf{A}_{i1}, \mathbf{A}_{i2}$) and ($\mathbf{B}_{i1}, \mathbf{B}_{i2}$)), forming a parallelogram (see Fig. 1). At the top, the arms are connected to the motors, while at the bottom, the forearms are connected to the nacelle. The latter is designed with four parts [8]: the two lateral bars ($[\mathbf{C}_1\mathbf{C}_2]$ and $[\mathbf{C}_3\mathbf{C}_4]$) and the two central bars linking lateral ones with revolute joints (Fig. 2). The nacelle also has an amplification system to transform the relative rotation θ into a proportional rotation ($\beta = \kappa\theta$) in the end-effector $\mathbf{E}$. While modeling the kinematics and dynamics, we assign $\mathbf{C}_4$ as the new end-effector position instead of its actual one for simplicity's sake and introduce the following notations:

This work is supported by the ANR-VIRAGO project.

The authors are with LASMEA-CNRS, Université Blaise Pascal, 24 Avenue des Landais 63177 Aubière Cedex, Clermont-Ferrand, France firstname.lastname@lasmea.univ-bpclermont.fr

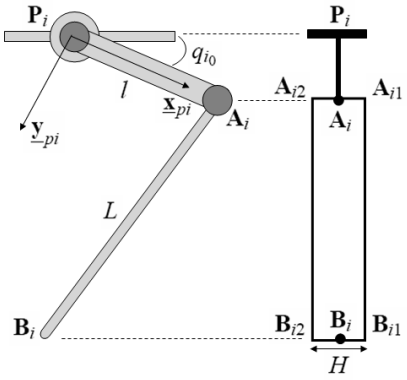

Fig. 1. Leg parameters.

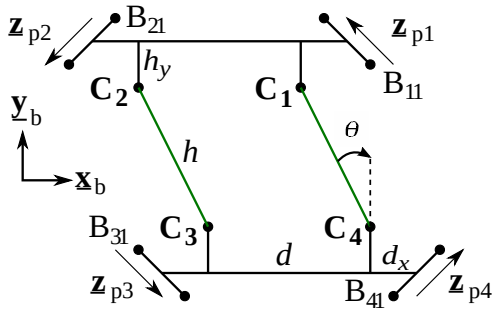

Fig. 2. Nacelle parameters.

- $i = 1,2,3,4$ denotes the legs and $j = 1,2$ denotes the edges of the forearm rods.
- Vectors are denoted with boldface characters and, in addition, unit vectors are underlined.
- $\mathcal{F}_b = (\mathbf{O}, \underline{\mathbf{x}}_b, \underline{\mathbf{y}}_b, \underline{\mathbf{z}}_b)$, $\mathcal{F}_e = (\mathbf{C_4}, \underline{\mathbf{x}}_e, \underline{\mathbf{y}}_e, \underline{\mathbf{z}}_e)$, $\mathcal{F}_c = (\mathbf{O_c}, \underline{\mathbf{x}}_c, \underline{\mathbf{y}}_c, \underline{\mathbf{z}}_c)$, $\mathcal{F}_{pi} = (\mathbf{P_i}, \underline{\mathbf{x}}_{pi}, \underline{\mathbf{y}}_{pi}, \underline{\mathbf{z}}_{pi})$ and $\mathcal{F}_{ai} = (\mathbf{A_i}, \underline{\mathbf{x}}_{ai}, \underline{\mathbf{y}}_{ai}, \underline{\mathbf{z}}_{ai})$ denote respectively the base, end-effector, camera and i^{th} arm and i^{th} forearm reference frames.
- All the parameters are expressed in $\mathcal{F}_c$.
- q_{i_0} is the articulated position of the i^{th} arm.
- In $\mathcal{F}_b$ the axes of the arm are designed as:

$$
\begin{aligned}
{}^b\underline{\mathbf{x}}_{pi} &= \begin{bmatrix} \cos(q_{i_0})\cos(\alpha_i) & \cos(q_{i_0})\sin(\alpha_i) & -\sin(q_{i_0}) \end{bmatrix}^T \\
{}^b\underline{\mathbf{z}}_{pi} &= \begin{bmatrix} -\sin(\alpha_i) & \cos(\alpha_i) & 0 \end{bmatrix}^T \\
{}^b\underline{\mathbf{y}}_{pi} &= {}^b\underline{\mathbf{z}}_{pi} \times {}^b\underline{\mathbf{x}}_{pi}
\end{aligned}
$$

  where $\alpha_i = \alpha + (i-1)\frac{\pi}{2}$.
- ${}^c\mathbf{V}_e = [\dot{x}, \dot{y}, \dot{z}]^T$ and ω_z are, in turn, the translational velocity and the angular velocity around the fixed axis ${}^c\underline{\mathbf{z}}_e$ of the end-effector $\mathbf{C_4}$. Thus, the Cartesian pose velocity of the end-effector can be represented by:

$$
{}^c\dot{\zeta} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} & \omega_z \end{bmatrix}^T
$$

- $\overrightarrow{{}^c\mathbf{A}_{ij}\mathbf{B}_{ij}} = \overrightarrow{{}^c\mathbf{A}_{i1}\mathbf{B}_{i1}} = \overrightarrow{{}^c\mathbf{A}_{i2}\mathbf{B}_{i2}} = L\,{}^c\underline{\mathbf{x}}_{ai}$.
- $\overrightarrow{{}^c\mathbf{P}_i\mathbf{A}_i} = l\,{}^c\underline{\mathbf{x}}_{pi}$.
- $\overrightarrow{{}^c\mathbf{A}_{i1}\mathbf{A}_{i2}} = \overrightarrow{{}^c\mathbf{B}_{i1}\mathbf{B}_{i2}} = H\,{}^c\underline{\mathbf{z}}_{pi}$.

III. VECTOR-BASED KINEMATICS

Here, some contexts necessary for the following sections are briefly recalled. For detailed explanations the reader is referred to [7].

A. Representation and Projection of a Leg

The rods $[\mathbf{A}_{i1}\mathbf{B}_{i1}]$ and $[\mathbf{A}_{i2}\mathbf{B}_{i2}]$ of the forearms are represented with binormalized Plücker coordinates $(\underline{\mathbf{x}}, \underline{\mathbf{n}}, n)$ [9]. Here $\underline{\mathbf{x}}$, $\underline{\mathbf{n}}$ and n denote the direction of the rod, the unit vector orthogonal to the plane defined by the rod and the center of projection, and the orthogonal distance of rod to the center of projection, respectively. One advantage of this representation is that $\underline{\mathbf{n}}$, meanwhile, corresponds to the image projection of the rod.

Assuming that the attachment point $\mathbf{B}_{i1}$ is lying on the revolution axis of the leg (see Fig. 3), the geometry of the robot calls forth the following constraints:

$$
{}^c\underline{\mathbf{n}}_i^{j\,T}\,{}^c\underline{\mathbf{x}}_{ai} = 0\,, \quad {}^c\mathbf{B}_{i1}{}^T\,{}^c\underline{\mathbf{n}}_i^j = -R\,, \quad {}^c\underline{\mathbf{x}}_{ai} = \frac{{}^c\underline{\mathbf{n}}_i^1 \times {}^c\underline{\mathbf{n}}_i^2}{\|{}^c\underline{\mathbf{n}}_i^1 \times {}^c\underline{\mathbf{n}}_i^2\|} \tag{1}
$$

where ${}^c\underline{\mathbf{x}}_{ai}$, ${}^c\underline{\mathbf{n}}_i^j$ and R are the direction, the edge and the radius of the cylindrical leg, respectively.

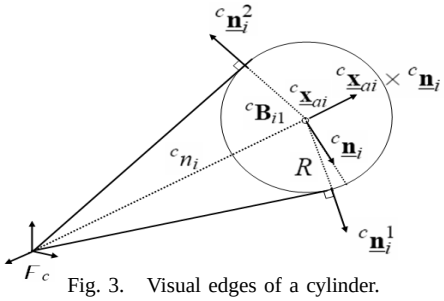

Fig. 3. Visual edges of a cylinder.

B. Inverse Differential Kinematic Models of The Robot

The implicit kinematic modeling through the observation of the first legs $[\mathbf{A}_{i1}\mathbf{B}_{i1}]$ is noted as:

$$
\dot{q}_{i_0} = {}^c\mathbf{D}_{e_i}^{inv}\,{}^c\dot{\zeta}\,, \qquad {}^c\dot{\underline{\mathbf{x}}}_{ai} = {}^c\mathbf{M}_i\,{}^c\dot{\zeta} \tag{2}
$$

whose expressions are algebraic and can be found in [7].

IV. VECTOR-BASED DYNAMICS

Here, it is demonstrated, step by step, how to obtain a simplified inverse dynamic model for the robot through the fusion of Khalil's [10] and Kane's methods [11]. The reasons for choosing these methods are their modularity and ability to be harmonized in a vector-based notation. The compactness of the model comes from the direct imposition of the leg orientations, granted by vision, into the equations of motion.

A. Inverse Dynamic Model of The Robot

Khalil's formulation [10] is followed to calculate the inverse dynamic model of the Quattro parallel robot:

$$
\Gamma = {}^c\mathbf{D}_e^T \left[\mathbb{F}_p + \sum_{i=1}^{4} \left({}^cJ_{\mathbf{B}_i}^T\,{}^cJ_i^{inv^T}\,H_i \right) \right] \tag{3}
$$

where ${}^c\mathbf{D}_e$ is the inverse of the inverse differential kinematic model ${}^c\mathbf{D}_e^{inv}$. The 4 d.o.f. platform dynamics ($\mathbb{F}_p \in \Re^{4\times 1}$) is computed via Newton-Euler formulation as in [10]. The relation between the Cartesian velocities of the terminal point of the i^{th} leg and the end-effector pose is,

$$
{}^cJ_{\mathbf{B}_i} = \begin{bmatrix} \mathbf{I}_3 & -\varepsilon_i h\,{}^c\underline{\mathbf{x}}_e \end{bmatrix} \tag{4}
$$

where $\varepsilon_1=\varepsilon_2=1,\ \varepsilon_3=\varepsilon_4=0$. ${}^cJ_i^{inv}\in\Re^{3\times3}$ and $H_i\in\Re^{3\times1}$ are the inverse differential kinematic and inverse dynamic models for the leg i, respectively. The following two subsections are devoted to their derivations.

B. Inverse Differential Kinematic Model of a Leg

Since the legs hold the $\underline{R}-(S-S)_2=\underline{R}-U-U$ structure equality [12], each of the legs of the robot is treated as 3 d.o.f. by omitting the joints that connect the leg to the moving platform. Respectively, 1 d.o.f. for the actuated revolute joint $\underline{R}$ and 2 d.o.f. for the passive universal joint U are required. After that, the instantaneous pose of the leg is specified with the *generalized coordinates* $\{q_{i_0},q_{i_1},q_{i_2}\}$ (Fig. 4), where q_{i_0} and $\{q_{i_1},q_{i_2}\}$ designate the radian measures of the angles of the arm and forearm orientations, respectively. To obtain the differential kinematic model of the leg, the terminal point position ${}^c\mathbf{B}_{i1}$ is written as below:

$$ {}^c\mathbf{B}_{i1}={}^c\mathbf{P}_i+l\,{}^c\underline{\mathbf{x}}_{pi}+\overrightarrow{{}^c\mathbf{A}_i\mathbf{A}_{i1}}+L\,{}^c\underline{\mathbf{x}}_{ai} \tag{5}$$

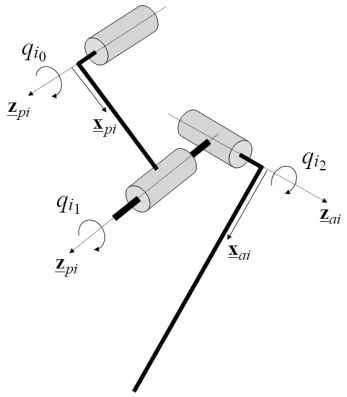

Fig. 4. Open tree structure ($\underline{R}-U-$) of the leg. ($\underline{\mathbf{z}}_{pi}\perp\underline{\mathbf{z}}_{ai}$)

Then, expressing the angular velocities of the kinematic chain with respect to fixed base frame yields:

$$ {}^c\omega_{pi}=\dot{q}_{i_0}\,{}^c\underline{\mathbf{z}}_{pi},\qquad {}^c\omega_{ai}=(\dot{q}_{i_0}+\dot{q}_{i_1})\,{}^c\underline{\mathbf{z}}_{pi}+\dot{q}_{i_2}\,{}^c\underline{\mathbf{z}}_{ai} \tag{6}$$

where ${}^c\omega_{pi}$ and ${}^c\omega_{ai}$ represent the angular velocities of the arm and forearm. Differentiating (5) yields:

$$ \frac{d}{dt}({}^c\mathbf{B}_{i1})=l\,{}^c\dot{\underline{\mathbf{x}}}_{pi}+L\,{}^c\dot{\underline{\mathbf{x}}}_{ai} \tag{7}$$

where

$$ {}^c\dot{\underline{\mathbf{x}}}_{pi}={}^c\omega_{pi}\times{}^c\underline{\mathbf{x}}_{pi},\qquad {}^c\dot{\underline{\mathbf{x}}}_{ai}={}^c\omega_{ai}\times{}^c\underline{\mathbf{x}}_{ai} \tag{8}$$

Equation (7), using (8), can be rewritten as below:

$$ {}^c\dot{\mathbf{B}}_{i1}=\underbrace{\begin{bmatrix}{}^c\mathbf{v}_i & s_i\,{}^c\underline{\mathbf{z}}_{ai} & s_i\,{}^c\underline{\mathbf{y}}_{ai}\end{bmatrix}}_{{}^cJ_i}\begin{bmatrix}\dot{q}_{i_0}\\ \dot{q}_{i_1}\\ \dot{q}_{i_2}\end{bmatrix} \tag{9}$$

where

$$ s_i=L\,\|{}^c\underline{\mathbf{z}}_{pi}\times{}^c\underline{\mathbf{x}}_{ai}\|,\qquad {}^c\mathbf{v}_i=l\,{}^c\underline{\mathbf{y}}_{pi}+\ s_i\,{}^c\underline{\mathbf{z}}_{ai} \tag{10}$$

and ${}^cJ_i\in\Re^{3\times3}$ is the forward differential kinematic model of the i^{th} leg. Working on the matrix ${}^cJ_i$, the analytic form of the inverse differential kinematic model of the leg is derived as follows:

$$ {}^cJ_i^{inv}=\frac{1}{s_i}\begin{bmatrix}\frac{s_i}{{}^c\mathbf{v}_i\cdot{}^c\underline{\mathbf{x}}_{ai}} & 0 & 0\\ \frac{-{}^c\mathbf{v}_i\cdot{}^c\underline{\mathbf{z}}_{ai}}{{}^c\mathbf{v}_i\cdot{}^c\underline{\mathbf{x}}_{ai}} & 0 & 1\\ \frac{-{}^c\mathbf{v}_i\cdot{}^c\underline{\mathbf{y}}_{ai}}{{}^c\mathbf{v}_i\cdot{}^c\underline{\mathbf{x}}_{ai}} & 1 & 0\end{bmatrix}\begin{bmatrix}{}^c\underline{\mathbf{x}}_{ai} & {}^c\underline{\mathbf{y}}_{ai} & {}^c\underline{\mathbf{z}}_{ai}\end{bmatrix}^T \tag{11}$$

C. Inverse Dynamic Model of a Leg

Here, Kane's method [11] is employed to obtain the inverse dynamic model H_i, which will be later plugged into (3) to complete the full inverse dynamic model. It is also the part where the leg orientations come into the picture in the dynamic model. In the sequel, a brief description of Kane's method is given and each step on the way to the computation of H_i is exhibited.

Let $\{F_{u_r}^*,F_{u_r}\}|_{r=1}^n$ be respectively the *generalized inertia forces* and *generalized active forces* for a system with n degrees of freedom, and defined as:

$$ F_{u_r}^*=\sum_{k=1}^{p}\left(\left(\frac{\partial\mathbf{v}_{c_k}}{\partial u_r}\right)^T\mathbf{F}_{in_k}+\left(\frac{\partial\omega_k}{\partial u_r}\right)^T\mathbf{T}_{in_k}\right),\quad r=1,\ldots,n \tag{12}$$

$$ F_{u_r}=\sum_{k=1}^{p}\left(\left(\frac{\partial\mathbf{v}_{c_k}}{\partial u_r}\right)^T\mathbf{F}_{k}+\left(\frac{\partial\omega_k}{\partial u_r}\right)^T\mathbf{T}_{k}\right),\qquad r=1,\ldots,n \tag{13}$$

where p is the number of rigid bodies, $u_r|_{r=1}^n$ are the *generalized speeds*, $\{\frac{\partial\mathbf{v}_{c_k}}{\partial u_r},\frac{\partial\omega_k}{\partial u_r}\}$ are partial linear and angular velocities, $\{\mathbf{F}_{in_k},\mathbf{T}_{in_k}\}$, and $\{\mathbf{F}_k,\mathbf{T}_k\}$ are the *inertia force/torque* generated by the accelerated masses and inertias, and *resultant force/torque* which is equivalent to a set of contact and distance forces acting on the k^{th} body, respectively. To have the equations of motion, namely Kane's dynamical equations, one just needs to add the generalized inertia and active forces and equate them to zero,

$$ F_{u_r}^*+F_{u_r}=0,\quad r=1,\ldots,n \tag{14}$$

1) Defining the partial velocities: Traditionally, the *generalized speeds* in Kane's method are defined as functions of the derivatives of *generalized coordinates*. In spite of that, we break out of the conventional routine by introducing the $\{\underline{\mathbf{x}}_{pi},\underline{\mathbf{x}}_{ai}\}$ as *generalized vectors* to express the instantaneous configuration of the leg, and define the following *generalized velocities*:

$$ \mathbf{u}_{i_1}\triangleq{}^c\dot{\underline{\mathbf{x}}}_{pi},\qquad \mathbf{u}_{i_2}\triangleq{}^c\dot{\underline{\mathbf{x}}}_{ai} \tag{15}$$

which make conceptually and visually simple to understand and to follow the equations of motion. Then, the velocities of the mass centers of the arm and forearm are computed as below:

$$ {}^c\mathbf{v}_{pi_c}=\tfrac{l}{2}\,{}^c\dot{\underline{\mathbf{x}}}_{pi},\qquad {}^c\mathbf{v}_{ai_c}=l\,{}^c\dot{\underline{\mathbf{x}}}_{pi}+\tfrac{L}{2}\,{}^c\dot{\underline{\mathbf{x}}}_{ai} \tag{16}$$

and, since ${}^c\underline{\mathbf{z}}_{pi}$ is orthogonal to ${}^c\underline{\mathbf{z}}_{ai}$, the angular velocities associated to the arm and forearm are resolved as follows:

$$ {}^c\omega_{pi}={}^c\underline{\mathbf{x}}_{pi}\times{}^c\dot{\underline{\mathbf{x}}}_{pi},\qquad {}^c\omega_{ai}={}^c\underline{\mathbf{x}}_{ai}\times{}^c\dot{\underline{\mathbf{x}}}_{ai} \tag{17}$$

Finally using (16) and (17), the partial linear and angular velocity tensors are tabulated as in Table I with respect to the generalized velocities in (15). In Table I, the $[\cdot]_x$ denotes a skew-symmetric matrix associated with a vector.

TABLE I

PARTIAL LINEAR AND ANGULAR VELOCITY TENSORS

	${}^c\mathbf{v}_{pi_c}$	${}^c\mathbf{v}_{ai_c}$	${}^c\omega_{pi}$	${}^c\omega_{ai}$
$\mathbf{u}_{i_1}$	$\frac{l}{2}\mathbf{I}_3$	$l\,\mathbf{I}_3$	$[{}^c\underline{\mathbf{x}}_{pi}]_x$	$\mathbf{0}_3$
$\mathbf{u}_{i_2}$	$\mathbf{0}_3$	$\frac{L}{2}\mathbf{I}_3$	$\mathbf{0}_3$	$[{}^c\underline{\mathbf{x}}_{ai}]_x$

2) Generalized inertia forces: The generalized inertia forces of the leg can be obtained in the light of (12) using the inertia forces ($\mathbf{F}_{in_k}$) and torques ($\mathbf{T}_{in_k}$) of the arm and forearm. To compute the inertia forces and torques from the Newton-Euler equations, the acceleration of mass centers are derived as follows,

$$ {}^c\mathbf{a}_{pi_c} = \tfrac{l}{2}\,{}^c\ddot{\underline{\mathbf{x}}}_{pi}\,, \qquad {}^c\mathbf{a}_{ai_c} = l\,{}^c\ddot{\underline{\mathbf{x}}}_{pi} + \tfrac{L}{2}\,{}^c\ddot{\underline{\mathbf{x}}}_{ai} \tag{18} $$

Respectively, by differentiating (17), the angular accelerations of the arm and forearm are derived as:

$$ {}^c\alpha_{pi} = {}^c\underline{\mathbf{x}}_{pi} \times {}^c\ddot{\underline{\mathbf{x}}}_{pi}\,, \qquad {}^c\alpha_{ai} = {}^c\underline{\mathbf{x}}_{ai} \times {}^c\ddot{\underline{\mathbf{x}}}_{ai} \tag{19} $$

3) Generalized active forces: The generalized active forces of the leg can be calculated using (13) through the torques $\{\mathbf{T}_{pi}, \mathbf{T}_{ai}\}$ being exerted and the gravitational forces $\{\mathbf{G}_{pi}, \mathbf{G}_{ai}\}$ acting on the arm and forearm, respectively:

$$ \begin{array}{l|l} \mathbf{T}_{pi} = (\Gamma_{i0} - I_{m_i}\ddot{q}_{i_0} - \Gamma_{i_f} - \Gamma_{i1})\,{}^c\underline{\mathbf{z}}_{pi} & \mathbf{G}_{pi} = -m_1 g\,{}^c\underline{\mathbf{z}}_b \\ \mathbf{T}_{ai} = \Gamma_{i1}\,{}^c\underline{\mathbf{z}}_{pi} + \Gamma_{i2}\,{}^c\underline{\mathbf{z}}_{ai} & \mathbf{G}_{ai} = -m_2 g\,{}^c\underline{\mathbf{z}}_b \end{array} \tag{20} $$

where I_{m_i} is the motor inertia and Γ_{i_f} is the friction term offering resistance on the actuated joint,

$$ \Gamma_{i_f} = f_{v_i}\,\dot{q}_{i_0} + f_{c_i}\,sign(\dot{q}_{i_0}) \tag{21} $$

with f_{v_i} viscous and f_{c_i} Coulomb friction coefficients. The m_1, m_2 and g are the mass of the arm, mass of the forearm and constant of gravity, respectively.

4) Inverse dynamic model: Finally, the inverse dynamic model of each leg of the Quattro robot can be computed by reassembling all the above equations into (14), which is only a matter of algebraic manipulation. In the next Section, it shall be shown that the inverse dynamic model of each leg can be expressed as a function of its forearm direction:

$$ \gamma_i = H_i({}^c\ddot{\underline{\mathbf{x}}}_{ai}, {}^c\dot{\underline{\mathbf{x}}}_{ai}, {}^c\underline{\mathbf{x}}_{ai}) \tag{22} $$

where $\gamma_i = [\Gamma_{i0}, \Gamma_{i1}, \Gamma_{i2}]^T$ is the required torque vector for the motion of the leg. Here, ideally, $\{\Gamma_{i1}, \Gamma_{i2}\}$ should be identically 0, since they correspond to passive universal joint.

V. VISION IN KINEMATICS AND DYNAMICS

In this section, the minimum variable sets necessary to derive kinematic and dynamic models of the robot are explored and shown to be fully computable only from visual information.

A. Required Variable Set for Kinematics

The inverse differential kinematic models ${}^c\mathbf{D}_e^{inv}$, ${}^c\mathbf{M}_i$ and ${}^cJ_i^{inv}$ depend on the following variables:

- $\underline{\mathbf{y}}_{pi}$ the perpendicular vectors to the arms.
- $\underline{\mathbf{x}}_e$ x-axis of the end-effector frame.
- $\underline{\mathbf{x}}_{ai}$ the directions of the forearms.

So, we need to compute ${}^c\underline{\mathbf{y}}_{pi} = {}^c\underline{\mathbf{z}}_{pi} \times {}^c\underline{\mathbf{x}}_{pi}$. Here ${}^c\underline{\mathbf{z}}_{pi}$ is constant and ${}^c\underline{\mathbf{x}}_{pi}$ can be expressed as follows:

$$ {}^c\underline{\mathbf{x}}_{pi} = \frac{1}{l}\left({}^c\mathbf{B}_{i1} - {}^c\mathbf{P}_i - \overrightarrow{{}^c\mathbf{A}_i\mathbf{A}_{i1}} - L\,{}^c\underline{\mathbf{x}}_{ai}\right) \tag{23} $$

where l, L, ${}^c\mathbf{P}_i$ and $\overrightarrow{{}^c\mathbf{A}_i\mathbf{A}_{i1}}$ are constant parameters while the directions of the forearms ${}^c\underline{\mathbf{x}}_{ai}$ can be measured by *vision*. Then, the only remaining parameter to be provided is ${}^c\mathbf{B}_{i1}$. To build the last variable ${}^c\underline{\mathbf{x}}_e = {}^c\underline{\mathbf{y}}_e \times {}^c\underline{\mathbf{z}}_e$ where ${}^c\underline{\mathbf{y}}_e = \frac{\overrightarrow{{}^c\mathbf{C}_4\mathbf{C}_1}}{h}$ and ${}^c\underline{\mathbf{z}}_e = {}^c\underline{\mathbf{z}}_b$, one only needs to know ${}^c\mathbf{C}_4$ and ${}^c\mathbf{C}_1$. Each of ${}^c\mathbf{C}_i$ can also be expressed again in terms of the ${}^c\mathbf{B}_{i1}$ and some known constant vectors and parameters:

$$ {}^c\mathbf{C}_4 = {}^c\mathbf{B}_{41} + \frac{H}{2}{}^c\underline{\mathbf{z}}_{p4} - d_x\,{}^c\underline{\mathbf{x}}_b + h_y\,{}^c\underline{\mathbf{y}}_b \tag{24} $$

$$ {}^c\mathbf{C}_1 = {}^c\mathbf{B}_{11} + \frac{H}{2}{}^c\underline{\mathbf{z}}_{p1} - d_x\,{}^c\underline{\mathbf{x}}_b - h_y\,{}^c\underline{\mathbf{y}}_b \tag{25} $$

Consequently, provided that the variables ${}^c\mathbf{B}_{i1}$, the attachment points of the legs onto the nacelle, are known, one can define all the required variable set for the kinematics. The computation of ${}^c\mathbf{B}_{i1}$ are explained in the next subsection.

B. Estimation of Attachment Points

Recalling the assumption that the attachment point ${}^c\mathbf{B}_{i1}$ of the rod on the travelling plate is lying on the revolution axis of the leg with radius R, the 2^{nd} constraint in (1) can be exploited by applying to both edges of the rods in legs 1 and 2, and yields:

$$ \begin{array}{ll} {}^c\underline{\mathbf{n}}_1^{1T}\,{}^c\mathbf{B}_{11} = -R & {}^c\underline{\mathbf{n}}_2^{1T}\,{}^c\mathbf{B}_{21} = -R \\ {}^c\underline{\mathbf{n}}_1^{2T}\,{}^c\mathbf{B}_{11} = -R & {}^c\underline{\mathbf{n}}_2^{2T}\,{}^c\mathbf{B}_{21} = -R \end{array} \tag{26} $$

Taking into account the travelling plate parameters, one can have the following relation:

$$ {}^c\mathbf{B}_{11} = {}^c\mathbf{B}_{21} + \frac{H}{2}{}^c\underline{\mathbf{z}}_{p2} + (d+2d_x)\,{}^c\underline{\mathbf{x}}_b - \frac{H}{2}{}^c\underline{\mathbf{z}}_{p1} \tag{27} $$

Replacing ${}^c\mathbf{B}_{11}$ in (26) with (27), the following linear system can be obtained from the image information:

$$ \begin{bmatrix} {}^c\underline{\mathbf{n}}_1^{1T} \\ {}^c\underline{\mathbf{n}}_1^{2T} \\ {}^c\underline{\mathbf{n}}_2^{1T} \\ {}^c\underline{\mathbf{n}}_2^{2T} \end{bmatrix} {}^c\mathbf{B}_{21} = \begin{bmatrix} -R - {}^c\underline{\mathbf{n}}_1^{1T}\left(\frac{H}{2}{}^c\underline{\mathbf{z}}_{p2} + (d+2d_x)\,{}^c\underline{\mathbf{x}}_b - \frac{H}{2}{}^c\underline{\mathbf{z}}_{p1}\right) \\ -R - {}^c\underline{\mathbf{n}}_1^{2T}\left(\frac{H}{2}{}^c\underline{\mathbf{z}}_{p2} + (d+2d_x)\,{}^c\underline{\mathbf{x}}_b - \frac{H}{2}{}^c\underline{\mathbf{z}}_{p1}\right) \\ \\ -R \\ -R \end{bmatrix} \tag{28} $$

The least-square solution, ${}^c\mathbf{B}_{21}$, of this 4×3 linear system is unique provided that 3 of the interpretation planes are linearly independent. Using (27), we can also arrive at ${}^c\mathbf{B}_{11}$.

After that, a second linear system can be rebuilt to compute ${}^c\mathbf{B}_{31}$ and ${}^c\mathbf{B}_{41}$ by repeating the same procedure on legs 3 and 4. We would like to point out that this estimation is performed in a single image. Note that this result was already presented in [13] on a real I4R robot, and is adapted here for the end-effector of the Quattro robot.

C. Required Variable Set for Dynamics

In addition to the required variable set of kinematics that was built in the above subsections, the inverse dynamic model of the robot depends on the following variables:

- $\underline{\mathbf{y}}_{ai}$, $\underline{\mathbf{z}}_{ai}$ the y and z axes of the forearm frames.
- $\underline{\ddot{\mathbf{x}}}_{pi}$, $\underline{\ddot{\mathbf{x}}}_{ai}$, $\underline{\dot{\mathbf{x}}}_{pi}$, $\underline{\dot{\mathbf{x}}}_{ai}$ accelerations and velocities of the unit vectors of the arm and forearm directions.
- $\ddot{q}_{i_0}$, $\dot{q}_{i_0}$ angular accelerations and velocities of the arms.

We compute the z and y axes of the forearm frames in the camera frame, using the *forearm directions*, as below:

$$ {}^c\underline{\mathbf{z}}_{ai} = ({}^c\underline{\mathbf{z}}_{pi} \times {}^c\underline{\mathbf{x}}_{ai})/\|{}^c\underline{\mathbf{z}}_{pi} \times {}^c\underline{\mathbf{x}}_{ai}\|, \quad {}^c\underline{\mathbf{y}}_{ai} = {}^c\underline{\mathbf{z}}_{ai} \times {}^c\underline{\mathbf{x}}_{ai} \tag{29} $$

The $\{{}^c\underline{\ddot{\mathbf{x}}}_{pi}, {}^c\underline{\dot{\mathbf{x}}}_{pi}\}$ can be computed by differentiating (23) with respect to time, which yields:

$$ {}^c\underline{\dot{\mathbf{x}}}_{pi} = \frac{1}{l}\left({}^c\dot{\mathbf{B}}_{i1} - L\,{}^c\underline{\dot{\mathbf{x}}}_{ai}\right), \quad {}^c\underline{\ddot{\mathbf{x}}}_{pi} = \frac{1}{l}\left({}^c\ddot{\mathbf{B}}_{i1} - L\,{}^c\underline{\ddot{\mathbf{x}}}_{ai}\right) \tag{30} $$

where $\{{}^c\ddot{\mathbf{B}}_{i1}, {}^c\dot{\mathbf{B}}_{i1}\}$ are obtained, using (4), as follows:

$$ {}^c\dot{\mathbf{B}}_{i1} = {}^cJ_{\mathbf{B}_i}\,{}^c\dot{\xi}, \quad {}^c\ddot{\mathbf{B}}_{i1} = {}^c\dot{J}_{\mathbf{B}_i}\,{}^c\dot{\xi} + {}^cJ_{\mathbf{B}_i}\,{}^c\ddot{\xi} \tag{31} $$

with ${}^c\dot{J}_{\mathbf{B}_i} = \begin{bmatrix} \mathbf{0}_{3\times 3} & -\varepsilon_i h\,\omega_z\,{}^c\underline{\mathbf{y}}_e \end{bmatrix}$. The velocities $\{\dot{q}_{i_0}, \underline{\dot{\mathbf{x}}}_{ai}\}$ can be handed in using the differential kinematic relations in (2), while the accelerations $\{\ddot{q}_{i_0}, \underline{\ddot{\mathbf{x}}}_{ai}\}$ have to be computed in two different ways whether the error is defined as a difference in the *Cartesian space* (CS) or in the *leg orientation space* (LS). In the case of a Cartesian error, the ${}^c\ddot{\xi}$ will come from the control law and accelerations will computed through:

$$ \ddot{q}_{i_0} = {}^c\dot{\mathbf{D}}_{e_i}^{inv}\,{}^c\dot{\xi} + {}^c\mathbf{D}_{e_i}^{inv}\,{}^c\ddot{\xi}, \quad {}^c\underline{\ddot{\mathbf{x}}}_{ai} = {}^c\dot{\mathbf{M}}_i\,{}^c\dot{\xi} + {}^c\mathbf{M}_i\,{}^c\ddot{\xi} \tag{32} $$

On the other hand, in the case of leg orientations error, the ${}^c\underline{\ddot{\mathbf{x}}}_{ai}$ will be coming from the control signal and this time the remaining acceleration $\ddot{q}_{i_0}$ will be obtained as follows:

$$ {}^c\ddot{\xi} = {}^c\mathbf{M}^{\dagger}({}^c\ddot{\mathbf{X}}_a - {}^c\dot{\mathbf{M}}\,{}^c\dot{\xi}), \quad \ddot{q}_{i_0} = {}^c\dot{\mathbf{D}}_{e_i}^{inv}\,{}^c\dot{\xi} + {}^c\mathbf{D}_{e_i}^{inv}\,{}^c\ddot{\xi} \tag{33} $$

where, in turn, ${}^c\mathbf{M} \in \Re^{12\times 4}$ and ${}^c\mathbf{X}_a \in \Re^{12\times 1}$ are the stacked matrices of ${}^c\mathbf{M}_i$ and ${}^c\underline{\mathbf{x}}_{ai}$. To compute (31) - (33), we need to know the pose velocity. The ${}^c\dot{\xi}$ can be either obtained by numerical differentiation of the pose or can be computed by differentiating the constraints in (26), assuming that vision can also quantify the edge velocities ${}^c\dot{\underline{\mathbf{n}}}_i^j$, and solving the linear systems for ${}^c\dot{\mathbf{B}}_{11}$ and ${}^c\dot{\mathbf{B}}_{41}$. To calculate ${}^c\dot{\mathbf{B}}_{11}$, the new linear system is written as follows:

$$ \begin{bmatrix} {}^c\underline{\mathbf{n}}_1^{1T} \\ {}^c\underline{\mathbf{n}}_1^{2T} \\ {}^c\underline{\mathbf{n}}_2^{1T} \\ {}^c\underline{\mathbf{n}}_2^{2T} \end{bmatrix} {}^c\dot{\mathbf{B}}_{11} = \begin{bmatrix} -{}^c\dot{\underline{\mathbf{n}}}_1^{1T}\,{}^c\mathbf{B}_{11} \\ -{}^c\dot{\underline{\mathbf{n}}}_1^{2T}\,{}^c\mathbf{B}_{11} \\ -{}^c\dot{\underline{\mathbf{n}}}_2^{1T}\,{}^c\mathbf{B}_{21} \\ -{}^c\dot{\underline{\mathbf{n}}}_2^{2T}\,{}^c\mathbf{B}_{21} \end{bmatrix} \tag{34} $$

while ${}^c\dot{\mathbf{B}}_{41}$ can be computed similarly. Then, the pose velocity can be expressed as below:

$$ {}^c\dot{\xi} = \begin{bmatrix} {}^c\dot{\mathbf{B}}_{41}{}^T & ({}^c\underline{\mathbf{y}}_e \times {}^c\dot{\underline{\mathbf{y}}}_e)\cdot {}^c\underline{\mathbf{z}}_e \end{bmatrix}^T \tag{35} $$

where ${}^c\dot{\mathbf{B}}_{41} = {}^c\dot{\mathbf{C}}_4$ and ${}^c\dot{\underline{\mathbf{y}}}_e = ({}^c\dot{\mathbf{B}}_{11} - {}^c\dot{\mathbf{B}}_{41})/h$.

Thereby, at this point *we substantiate that exploiting only the vision, it is possible to figure out the whole variable sets of kinematics and dynamics.* Note that this confluence is made easy, thanks to the vector-based formulation of both the dynamics and the differential geometry in the image.

VI. RESULTS

The proposed inverse dynamic model was verified with the simplified model proposed in [4], which has already proved to be as correct as the complete dynamic model of the Quattro prototype obtained on Adams software. In comparison of models, the maximum error rate of torques is found to be 4.84%, which means the modeling is accurate enough to be used in control. The direct dynamic model, that is used in the simulator, is also derived from [4], which brings on a certain level of disturbance directly to the control signal, since it is not the direct inverse of the proposed model.

The trajectory tracking simulations are conducted with a classical CTC at $500\,Hz$ expressed either in the CS or in the LS (Figs. 5 and 6). An Adept motion with 25 *mm* altitude and 300 *mm* length is chosen for performance evaluations in a pick-and-place task. The maximum velocity and acceleration of the motion are set as 1.34 m/s and $1G$, respectively. The simulations are executed with different noise levels, and results are compared. We injected either $10\,\mu m$ and 1% or $100\,\mu m$ and 10% uncertainty on the geometric and dynamic parameters, respectively. The sensor feedback measurements are also corrupted. In *Cartesian* space CTC, the feedback pose is perturbed with $\{10\,\mu m, 0.5^\circ\}$, $\{50\,\mu m, 1^\circ\}$ and $\{100\,\mu m, 2^\circ\}$, and in *leg orientation* space CTC, the feedback signals (leg orientation unit vectors) are independently deflected with 0.01°, 0.05° and 0.1° (degree), respectively. In fact, the injected noises in two spaces are not tenably comparable, since they are added at the final stage of the feedback signals, but defensively to give an intuition, the deflection of 0.1° in a leg orientation drifts the moving platform approximately $1.4\,mm$ away. The accuracy and precision of the performed motion are assessed in terms of mean and standard deviation of the tracking errors in Cartesian space. Figs. 7 and 8 depict the superimposed trajectories and tracking errors for Adept motion in both spaces without any noise. The accuracy and precision values are 2769 μm and 2183 μm for CS-CTC, and 137 μm and 102 μm for LS-CTC. Comparing the space trajectory (top) and the time trajectory (bottom) in Fig. 7, one can observe that the errors in x and z are essentially due to some delay in the tracking but do not appear as a deviation in space. In Tables II and III, the accuracy (bold) and precision values are tabulated for the same motion repeated under various noise levels. From the results, it seems that sensing the leg

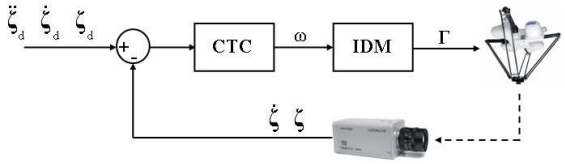

Fig. 5. Block diagram for Cartesian space CTC (CS-CTC).

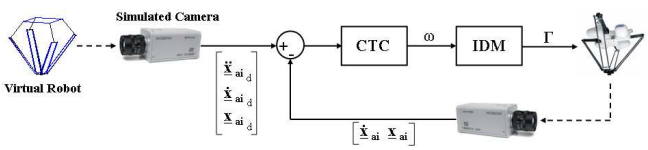

Fig. 6. Block diagram for leg orientation space CTC (LS-CTC).

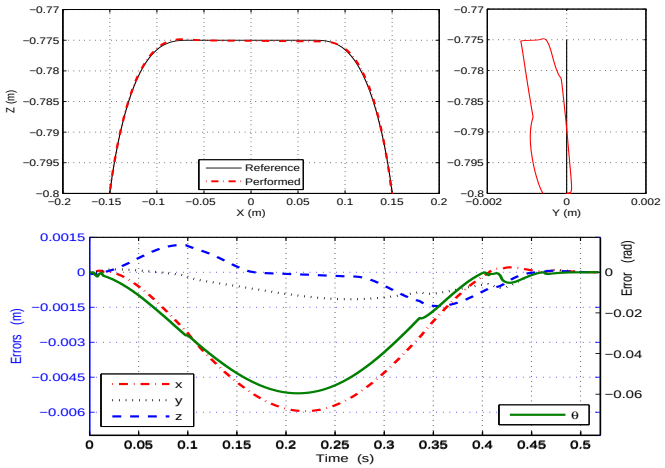

Fig. 7. Superimposed trajectories (top) and tracking errors in CS for CS-CTC. (tracking accuracy 2769 μm, precision 2183 μm) ($K_p = 10, K_i = 0, K_d = 85$)

orientations is more robust to errors since it is closer to the essential variables in modeling. Hence, it puts the observation of the end-effector out of being ultimate goal.

TABLE II

CARTESIAN SPACE TRACKING ERRORS (μm), FOR CS-CTC.

		Sensor (μm, deg)		
	Noise	10, 0.5°	50, 1°	100, 2°
Geo. (μm)	10	**3234**	**3648**	**4389**
Dyn. (%)	1	1828	1647	1717
Geo. (μm)	100	**3791**	**3860**	**4916**
Dyn. (%)	10	2172	1982	1697

TABLE III

CARTESIAN SPACE TRACKING ERRORS (μm), FOR LS-CTC.

		Sensor (deg)		
	Noise	0.01°	0.05°	0.1°
Geo. (μm)	10	**216**	**774**	**1500**
Dyn. (%)	1	77	172	287
Geo. (μm)	100	**235**	**805**	**1539**
Dyn. (%)	10	125	133	242

VII. CONCLUSIONS AND REMARKS

A novel vector-based approach for dynamic modeling and control of a Quattro robot, based on leg observations, is introduced and the first promising results of this new methodology are presented, which encourage us to put it in practice on a real Quattro robot. We will also define the error in image space to make the control more robust to errors and noises. In this work, all the required feedback information is provided solely by the vision sensor and the full control of the parallel robot is fulfilled only through the forearms' edges and their velocities in the image. In fact, the edge velocities of a forearm assumed that can be quantified from an image by vision, whereas they are numerically differentiated in simulations. So, we put forward for ourselves another objective to calculate them theoretically as well. But initially, we should dispose of the problem of detecting the leg edges in real images at high-speed.

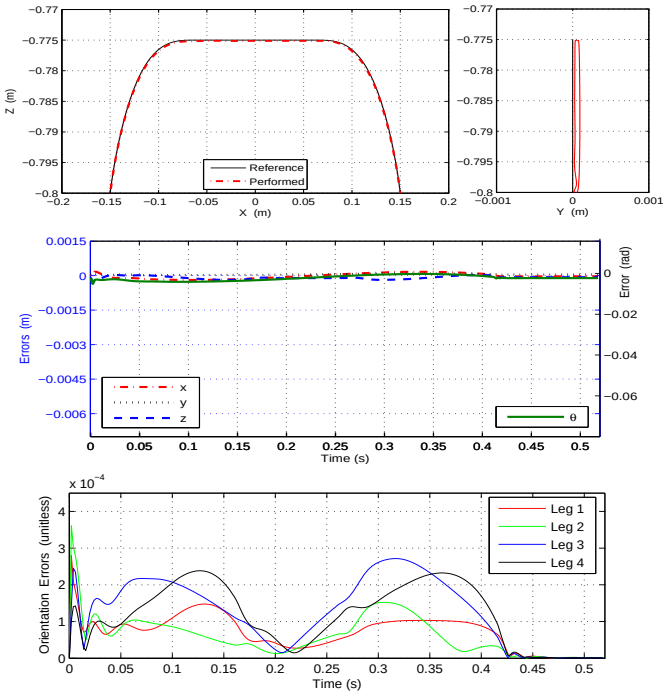

Fig. 8. Superimposed trajectories (top) and tracking errors in CS and LS ($\tilde{\mathbf{x}}_{ai}^T \tilde{\mathbf{x}}_{ai}$) for LS-CTC. (tracking accuracy 137 μm, precision 102 μm) ($K_p = 20000, K_i = 20000, K_d = 80$)

Consequently, once these snags along the way are dispelled, this work will induce a new way of controlling parallel mechanisms, since many of them contain slender structures.

REFERENCES

[1] Jean P. Merlet, *Parallel Robots*, Kluwer Academic Publishers, 2000.

[2] W. Khalil, E. Dombre, "Modeling, identification and control of robots", London, 2002.

[3] A. Vivas, P. Poignet, F. Pierrot, "Predictive functional control of a parallel robot", *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'03*, pp 2785-2790, Las Vegas, USA, October 2003.

[4] V. Nabat, S. Krut, O. Company, P. Poignet, F. Pierrot, "On the Design of a Fast Parallel Robot Based on Its Dynamic Model", *Springer-Verlag Berlin Heidelberg, Experimental Robotics*, January 30, 2008.

[5] L. Baron, J. Angeles, "The on-line direct kinematics of parallel manipulators under joint-sensor redundancy", *Advances in Robot Kinematics*, pp 126-137, Kluwer, 1998.

[6] N. Andreff, F. Martinet, "Kinematic modeling of some parallel manipulators for vision based control purposes", *The first European Conf. on Mechanism Science, EuCoMes*, Obergurg, Austria, 2006.

[7] T. Dallej, N. Andreff and F. Martinet, "Visual Servoing of Par4 using Leg Observation", *The 32nd Annual Conf. of the IEEE Industrial Electronics Society, (IECON'06)*, pp 3782-3787, Paris, France, 2006.

[8] V. Nabat, M. O. Rodrigues, O. Company, S. Kurt, F. Pierrot, "Par4: very high speed parallel robot for pick-and-place", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'05*, Alberta, Canada, 2005.

[9] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines", *Int. Journal of Robotics Research*, 21(8), pp 679-700, 2002.

[10] W. Khalil, O. Ibrahim, "General Solution for the Dynamic Modeling of Parallel Robots", *IEEE Int. Conf. on Robotics and Automation, (ICRA'04)*, New Orleans, LA, 2004.

[11] Kane, T.R., Levinson, D.A., "Dynamics: Theory and Applications", *McGraw Hill* New York, 1985.

[12] F. Pierrot, F. Marquet, O. Company, T. Gil, "H4 Parallel Robot: Modeling, Design and Preliminary Experiments", *IEEE Int. Conf. on Robotics and Automation, (ICRA'01)*, Seoul, Korea, 2001.

[13] T. Dallej, N. Andreff and F. Martinet, "Image-Based Visual Servoing of the I4R parallel robot without Proprioceptive Sensors", *IEEE Int. Conf. on Robotics and Automation, (ICRA'04)*, Roma, Italy, 2007.