Linear Dynamic Modeling of Parallel Kinematic Manipulators from Observable Kinematic Elements

Erol Özgür[a,*], Nicolas Andreff[b,a], Philippe Martinet[c,a]

[a]*Institut Pascal, Clermont Universités, Clermont-Ferrand, France*
[b]*Institut FEMTO-ST, Université de Franche-Comté, Besançon, France*
[c]*IRCCyN, Ecole Centrale de Nantes, Nantes, France*

Abstract

This paper presents a linear method for kinematic and dynamic modeling of parallel kinematic manipulators. This method is simple, compact and clear. One can write all the equations from the beginning till the end with pen and paper. It is thus well suited to mechanical understanding and computer implementation. We can apply it to many parallel robots. This method relies on a body-oriented representation of observable rectilinear kinematic structures (kinematic elements) which form the robot legs.

Keywords: Parallel kinematic manipulators, Body-oriented inverse dynamic model, Computer vision, Kinematic element, Leg observation

1. Introduction

Modeling of robots can be categorized in two groups: *application-oriented* methods and *analysis-oriented* methods. In application-oriented modeling the generic scenario is as follows: since the motor encoders that measure the articular positions are directly implanted in robots and are rich enough to supply the full configuration of the serial robots, they are immediately adopted as a basic medium for sensing.

This easily misleads one into also using them for parallel robots (*aka.* parallel kinematic manipulators). Indeed, when parallel robots are considered, the information delivered by motor encoders becomes poor because of

---

*Corresponding author — email: erol.ozgur@ifma.fr, tel: +33603375223

many other sensorless passive joints. If one expresses models with this insufficient active joint coordinates, then the models inflate, become slow, hard to understand and to implement. This inevitably urges one to offer simplifications [1] and to omit some of the modeling errors, thus giving simplified and fast [2] but less accurate new models for control. Thus, application-oriented methods, based on joint sensing, become inefficient when the structural complexity of the robot increases.

On the other hand, *analysis-oriented* methods mostly concentrate on finding efficient, intuitive, simple and linear procedures for synthesis and analysis of complex robots. Some of these analysis-oriented methods use Screw theory [3], [4], [5], [6], [7], [8], [9], [10] and Grassmann-Cayley [11] algebra which are based on lines of motions. These lines of motions are the *joint axes*, and these *analysis-oriented* methods model the joint motions. As a matter of fact, their practical applicability on real complex robots is limited. Instead, they are used for analysis.

Obviously, in these scenarios, the difficulties in modeling and applicability are sourced from the lack of an appropriate sensing. What if we had extra sensor(s)? What if we knew everything about the mechanism? This is the purpose of redundant metrology paradigm, introduced by [12] which turned the non-linear forward kinematic problem into a simple linear one. This approach was eventually used in [13] to propose a kinematic modeling method merging line geometry and the projection of physical edges of the robot structure into lines in images taken by a video camera. It was later used [14] to servo visually parallel robots by observing the legs of the robot with a camera rather than by reading the motor encoders or imaging the moving platform. Actually, the redundant metrology paradigm is a partial answer to the fact that most of the proposed approaches for deriving the kinematics and dynamics of a parallel robot suffer from the lack of efficiency, namely *"relative simplicity, ease of manipulation for purposes of designing automatic control systems and minimal consumption of time during numerical solution"* as defined by Kane *et al.* [15].

The method we propose here assumes, at first, that there is no restriction on the variable set selection needed for modeling, and suggests that the selected state variables should fulfil as much as possible the following criteria:

- *Algebraicity*: ability to be solved with formal tools, *e.g.*, linear algebra;
- *Completeness*: ability to represent fully both kinematics and dynamics;
- *Sensibility*: ability to be perceived directly by physical sensors;

2

- *Readability*: ability to allow for good and easy understanding;
- *Codability*: ability to be implemented easily on a computer;

so that the *simplicity*, *geometric intuitiveness* and *numerical efficiency* requested by Kane can be kept on the final expressions.

Thus, we came up with an original body-based modeling method rather than a joint-based. This body-based method contributes to modeling of parallel robots as follows:

- It uses lines to model concrete moving bodies rather than abstract joint axes. This enhances the visual perception of a robot, such that a human brain can imagine almost vividly a robot's motion by just reading the equations.
- It calculates the equations of motion with simple linear vector algebra and in a compact form. This eases the codability and the fast solvability of equations on the computer. Even for the most complex robots, the inverse dynamic model can be worked out with pen and paper.

The remainder of this paper proceeds as follows: Section 2 discusses various modeling approaches; Section 3 introduces body-oriented representation of parallel robots; Section 4 presents both kinematics and dynamics of a kinematic element; Then, Section 5 develops kinematic and dynamic models of a parallel robot; Sections 6 and 7 evaluate the presented modeling method from global and computational complexity points of view; Finally, Section 8 concludes the paper and proposes some future works.

2. Discussion on the Inspiring Works

In this section, Kane's, Khalil's, and Tsai's methods are discussed, because they have already made important steps toward improved efficiency and thus form a solid ground for the proposed method.

2.1. Khalil's Method

Khalil [16] proposed an inverse dynamic modeling method which is specific to parallel robots. This method expresses the dynamics of a parallel robot from the equilibrium of all forces applied on the moving platform, and it is decomposed into the following steps:

1. to consider each kinematic leg of a parallel robot as an *independent* serial robot;

2. to write the inverse dynamics of each kinematic leg using all the passive and active joint coordinates (redundant set of variables);

3. to transfer all the efforts of kinematic legs to the moving platform using their inverse velocity kinematic models;

4. finally, to sum all the forces collected on the moving platform and then to project the final total effort onto the active joints.

However, the strong drawback of this method is its loss of efficiency, because it requires sensing and actuation to be collocated. Moreover, the computation of the dynamics of each kinematic leg loses its intuitiveness, and it needs computation of the inverse of the forward velocity kinematic model of a serial kinematic leg because of the balance of all the efforts on the moving platform.

2.2. Tsai's Method

In [10], Tsai formulated the dynamics of parallel robots based on the virtual work principle. This formulation follows the steps below:

1. compute the kinematic twist and wrench twosome at the mass center of every link;

2. compute the link Jacobians relating the link kinematic twists to the end-effector kinematic twist.

3. finally, express all the virtual works of links, of actuators and of platform, in the end-effector frame through the computed Jacobians.

Expressing every effort in the end-effector frame with a similar way to Khalil, Tsai easily takes into account the motion constraints of the closed-loop kinematics. Using virtual work also simplifies the derivation of the final equations of motion. For application purposes, by MATLAB simulations, Tsai demonstrated his formulation for the Gough-Stewart platform where the equations were written based on joint values. However, as stated earlier, for real applications of parallel robots joint sensing is not enough. Since Tsai's work was rather for analysis purposes, he did not put any discussion for the applicability of his method even though it looks very intuitive.

2.3. Kane's Method

Kane's method was rarely addressed to parallel kinematics manipulators [17] [18] but its quest of efficiency is totally in line with the needs of parallel robot control and has strongly inspired this work.

With regards to efficiency, Kane has revealed the notion of *generalized speeds* to increase the efficiency of expressions [19]. The *generalized speeds* are functions of generalized coordinates (scalar joint values) and their speeds. Their choice is completely arbitrary, and is usually determined by inspection of the velocities of bodies in a mechanism. A good choice of *generalized speeds* can have important effect on the resulting equations. Once their choice is made, then the method proposed by Kane is simply a matter of:

1. derivation of the mechanism's kinematics in terms of the generalized coordinates and generalized speeds;
2. computation of all the forces that exist in the mechanism;
3. projection of these forces on the directions of motion (*i.e.*, directions associated to the generalized speeds) to obtain the generalized forces;
4. formal calculus (automatic generation of the model equations).

Expressed from the control point of view, choosing appropriate generalized speeds is a matter of defining the most appropriate dynamic state variables for a parallel robot. It now seems certain that using the actuator positions (independent set of generalized coordinates) as the static state variables is not necessarily the optimal choice, because there is usually not a single solution to the forward kinematic problem. Consequently, using the actuator positions and their velocities as the dynamic state variables is certainly not the appropriate choice either.

Since the inverse kinematic problem is usually well posed for a parallel robot, a more efficient choice is to use the end-effector Cartesian pose and its velocity as dynamic state variables [20]. This choice leads to efficient models *as long as one is able to measure or estimate the pose and velocity of the end-effector in the Cartesian space.* For instance, it was shown that such a choice makes Khalil's methodology totally linear since it replaces the non-linear computation of the forward kinematics of each leg [20]. However, there remains another loss of efficiency: one can not directly measure in the Cartesian space, so one has to estimate such variables, either through a mechanism or by optical means (*e.g.*, laser, vision). The estimation is always a non-linear problem, except (perhaps) for high cost laser trackers.

So, one question arises: what would be an efficient choice for the dynamic state variables ?

3. Body-Oriented Representation of Parallel Mechanisms

Parallel mechanisms are considered as an arrangement of joints subsidiarily connected by links, in continuity with the perspective of serial mechanisms where joints play an essential role. However, we believe that the serial/parallel duality [21],[22] should also hold for the description of the mechanisms. A reason for this is that most of the kinematic understanding of parallel mechanisms lies in geometry of the lines connecting successive joint centers in the mechanism (see for instance, the geometric interpretation of the singularities of a 3-RRR planar mechanism). Therefore, parallel mechanisms should be considered as an arrangement of links, from now on called *kinematic elements* due to their high kinematic value, subsidiarily connected by joints. This is also coherent with the intuitive description of a *parallel robot* as composed of a base platform, k kinematic legs and either an articulated nacelle or a moving rigid platform:

$$(parallel\ robot) : \left\{ (base\ platform), \begin{bmatrix} (kinematic\ leg)_1 \\ \vdots \\ (kinematic\ leg)_k \end{bmatrix}, \begin{pmatrix} nacelle\ or \\ moving\ platform \end{pmatrix} \right\}$$

(1)

where each component of this grammar can be made up from *kinematic elements*. The base platform is rigid and fixed, thus it will not be taken into account in modeling.

3.1. Definition of a Kinematic Element

In [23], Dallej has employed the "kinematic element" terminology for the last link of a kinematic chain of a parallel robot to propose a framework for the modeling and control of parallel robots at kinematic level. However, the notion of a kinematic element was never fully formalized. On the other hand, here, we give a mathematical model of a *kinematic element* which is intuitive and empiric. Furthermore, we are interested not only with the kinematics but also with the dynamics of these *kinematic elements*.

Remark: In IFToMM dictionary (standardization of terminology of mechanisms and machine science), the term "element" has already been defined as a solid body or a fluid component of a mechanism [24]. Here, we augmented this term with the adjective "kinematic" to express our new *elementary structure* (*i.e.*, rectilinear kinematic sub-chain) which has various mobilities. For the time being, we could not find a better name.
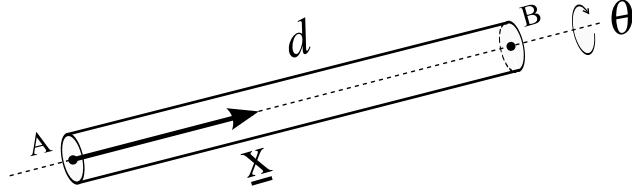
Figure 1: A kinematic element with its input joint center **A**, unit 3D direction vector $\underline{\mathbf{x}}$, length d and rotation angle θ around the direction vector. **B** is the output joint center.

3.1.1. Geometric Representation of a Kinematic Element

Here, we propose a model where a kinematic element can have *extrinsic* and *intrinsic* mobilities. The extrinsic mobility specifies the mobility of the 3D line supporting the physical kinematic element (*e.g.*, the axis of a cylindrical body) and is thus defined as:

$$(extrinsic\ mobility\ state): \{\, \mathbf{A},\, \underline{\mathbf{x}}\, \} \tag{2}$$

where **A** is the input point of the kinematic element, and $\underline{\mathbf{x}}$ is the unit $3D$ direction vector of the 3D line. This extrinsic mobility is sourced from the location of the previous kinematic element in the leg which changes **A**; and from a revolute or universal or spherical joint located in **A** which changes $\underline{\mathbf{x}}$.

The intrinsic mobility specifies the mobility of the kinematic element along and around its supporting 3D line:

$$(intrinsic\ mobility\ state): \{\, d,\, \theta\, \} \tag{3}$$

where d and θ respectively denote the length along $\underline{\mathbf{x}}$ and angle around $\underline{\mathbf{x}}$. This intrinsic mobility is sourced from a prismatic joint which changes d and which is located inside the kinematic element; and from a revolute joint which changes θ and which is located inside the kinematic element; or from a spherical joint which changes θ and which is located in **A**. Thus, a generic geometric representation of a kinematic element (see Fig. 1) can be described as follows:

$$(geometric\ state): \{\, \mathbf{A},\, \underline{\mathbf{x}},\, d,\, \theta\, \} \tag{4}$$

Remark: A spherical or universal joint is decomposed into two parts: one part which rotates the kinematic element around its axis, and the other part which rotates the axis of the kinematic element. So, the first part is associated to intrinsic mobility of the kinematic element, and the second part to extrinsic mobility.

7

Analogy. *Chasles' theorem [25] states that any rigid body displacement can be reduced to a canonical form, where this displacement is achieved by a rotation (θ) around a line L and a translation (d) along the same line L.*

This implies that a kinematic element, which uses the same geometric variables for its state (see Fig. 1), can be considered as a physical (visually concrete) representation of the canonic displacement between its connection points with the previous and next kinematic elements in the leg.

Distinguishing between intrinsic and extrinsic mobilities allows us to define some basic kinematic element types such as:

Bar Element $[B]$ It has only extrinsic mobility, and it is defined by:

$$(bar\ state): \underbrace{\{\,\mathbf{A},\,\underline{\mathbf{x}}\,\}}_{varying} \cup \underbrace{\{\,d,\,\theta\,\}}_{constant} \tag{5}$$

Actually, the B-element is the most common type and exists almost in every parallel robot, *e.g.*, parallelogram kinematic structures.

Spindle Element $[Sp]$ It is a B-element which can be also rotated around its axis. Thus, we write its geometric state as follows:

$$(spindle\ state): \underbrace{\{\,\mathbf{A},\,\underline{\mathbf{x}},\,\theta\,\}}_{varying} \cup \underbrace{\{\,d\,\}}_{constant} \tag{6}$$

Zlatanov's 3-URU DYMO parallel mechanism has some Sp-elements [26].

Telescopic Element $[T]$ It is a B-element which can be also elongated along its axis. Its state is thus given by:

$$(telescopic\ state): \underbrace{\{\,\mathbf{A},\,\underline{\mathbf{x}},\,d\,\}}_{varying} \cup \underbrace{\{\,\theta\,\}}_{constant} \tag{7}$$

The struts in a Gough-Stewart manipulator are the T-elements.

Screw Element $[S]$ It is the most generic one and has full mobility:

$$(screw\ state): \underbrace{\{\,\mathbf{A},\,\underline{\mathbf{x}},\,d,\,\theta\,\}}_{varying} \tag{8}$$

The fourth leg of the Delta parallel robot (*i.e.*, the one that gives a rotation to the end-effector) is an example of such an S-element.

Remark: Point $\mathbf{A}$ becomes constant if it is attached to the base platform.

3.1.2. Dynamic Representation of a Kinematic Element

We define the dynamic representation of a kinematic element as follows:

$$(\textit{dynamic state}) : \{\, \mathbf{A}, \underline{\mathbf{x}}, d, \theta \,\} \;\cup\; \{\, m, \mathcal{I}, \mathbf{S}, \dot{\mathbf{S}}, \ddot{\mathbf{S}}, \dot{\mathbf{A}}, \dot{\underline{\mathbf{x}}}, \dot{d}, \dot{\theta}, \ddot{\mathbf{A}}, \ddot{\underline{\mathbf{x}}}, \ddot{d}, \ddot{\theta} \,\} \;\cup$$
$$\{\, \mathbf{f_g}, \mathbf{f}^*, \boldsymbol{\tau}^*, \boldsymbol{\tau}_{\underline{\mathbf{x}}}, \boldsymbol{\tau}_{\underline{\mathbf{x}}}^*, \bar{\boldsymbol{\tau}}_{\underline{\mathbf{x}}}, \mathbf{f}_d, \mathbf{f}_d^*, \bar{\mathbf{f}}_d, \boldsymbol{\tau}_\theta, \boldsymbol{\tau}_\theta^*, \bar{\boldsymbol{\tau}}_\theta \,\}$$

$$(9)$$

which contains intrinsic dynamic parameters and higher-order kinematics:

- its mass m and its central inertia dyadic $\mathcal{I}$;
- its mass center position, velocity and acceleration: $\mathbf{S}$, $\dot{\mathbf{S}}$ and $\ddot{\mathbf{S}}$;
- the velocities of its geometric state variables: $\dot{\mathbf{A}}$, $\dot{\underline{\mathbf{x}}}$, $\dot{d}$, $\dot{\theta}$;
- the accelerations of its geometric state variables: $\ddot{\mathbf{A}}$, $\ddot{\underline{\mathbf{x}}}$, $\ddot{d}$, $\ddot{\theta}$;

and as well as forces and torques:

- its gravity force $\mathbf{f_g}$;
- its body inertial force $\mathbf{f}^*$ and inertial torque $\boldsymbol{\tau}^*$;
- an active extrinsic torque $\boldsymbol{\tau}_{\underline{\mathbf{x}}}$ of an extrinsic rotary actuator that turns the kinematic element around $\underline{\mathbf{z}}$ axis of its body frame, where

$$\underline{\mathbf{y}} = \dot{\underline{\mathbf{x}}} / \|\dot{\underline{\mathbf{x}}}\|, \qquad (\dot{\underline{\mathbf{x}}} \perp \underline{\mathbf{x}}), \qquad \underline{\mathbf{z}} = \underline{\mathbf{x}} \times \underline{\mathbf{y}} \qquad (10)$$

  and as well as the inertial torque $\boldsymbol{\tau}_{\underline{\mathbf{x}}}^*$ and the frictional torque $\bar{\boldsymbol{\tau}}_{\underline{\mathbf{x}}}$ of this rotary actuator;
- an active intrinsic force $\mathbf{f}_d$ which elongates or shortens the kinematic element along its direction $\underline{\mathbf{x}}$, and as well as the inertial force $\mathbf{f}_d^*$ and frictional force $\bar{\mathbf{f}}_d$ of this active prismatic joint;
- an active intrinsic torque $\boldsymbol{\tau}_\theta$ of an intrinsic rotary actuator that turns the kinematic element around its direction $\underline{\mathbf{x}}$, and as well as the inertial torque $\boldsymbol{\tau}_\theta^*$ and frictional torque $\bar{\boldsymbol{\tau}}_\theta$ of this intrinsic rotary actuator;

3.2. Representation of a Moving Platform and a Nacelle

The moving platform is a single rigid body, and the nacelle is a set of articulated rigid bodies where one of them is the moving platform. In practice, we split a rigid moving platform into exactly k kinematic elements, and an articulated nacelle into nk kinematic elements ($n \in \{1, 2, \ldots\}$), so that each kinematic leg of the mechanism can be augmented equally with the corresponding split part(s) of the moving platform or the nacelle. In Fig. 2, for example, the rigid moving platform of the Gough-Stewart mechanism (*resp.*

the nacelle of the H4 parallel robot [27]) is split into 6 (*resp.* 8) kinematic elements such that they share equally both the mass and the inertia of the moving platform (*resp.* the parts of the nacelle that they belong to). Mass centers of the kinematic elements are concentric (black dots) and located at the same position (big circles containing black dots) with the real mass center(s) of the moving platform of the Gough-Stewart mechanism and the parts of the nacelle of the H4 parallel robot.
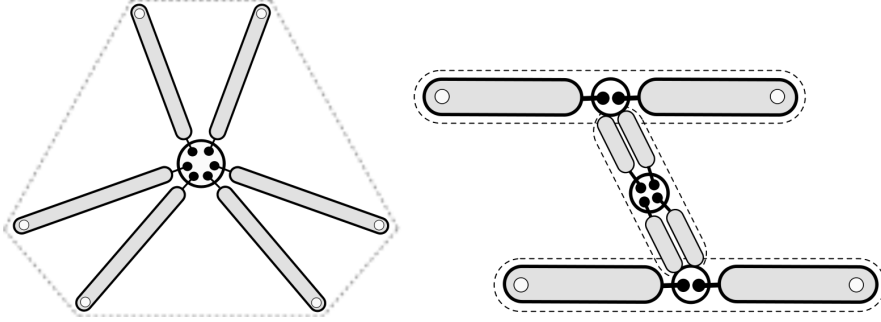

Figure 2: The moving platform of the Gough-Stewart mechanism is split into 6 kinematic elements (left) and the nacelle of the H4 parallel robot is split into 8 kinematic elements.

4. Kinematic and Dynamic Model of a Kinematic Element

4.1. Kinematics of a Kinematic Element

Positions: The output articulation point (the end point) $\mathbf{B}$ and the mass center $\mathbf{S}$ of a kinematic element can be computed as follows:

$$\mathbf{B} = \mathbf{A} + d\underline{\mathbf{x}}, \qquad \mathbf{S} = \mathbf{A} + x\underline{\mathbf{x}} + \mathbf{e} \qquad (11)$$

where x is the projection coordinate of the mass center onto the direction vector $\underline{\mathbf{x}}$ of the kinematic element, and $\mathbf{e}$ is a vector representing the eccentricity of the mass center to the direction axis $\underline{\mathbf{x}}$ of the kinematic element ($\mathbf{e} \perp \underline{\mathbf{x}}$). If the kinematic element is axis-symmetric and it has a uniform mass distribution along $\underline{\mathbf{x}}$, then $x = d/2$ and $\mathbf{e} = \mathbf{0}$.

Rotational Velocity and Acceleration:

Lemma. (*proof in [28]*) *Rotational velocity of a kinematic element, expressed in a fixed reference frame with respect to the same fixed reference frame, can be directly written with its unit direction vector, the velocity of it and the angular velocity of a kinematic element around its unit direction vector:*

$$\boldsymbol{\omega} = \underline{\mathbf{x}} \times \underline{\dot{\mathbf{x}}} + \dot{\theta}\,\underline{\mathbf{x}} \qquad (12)$$

From this Lemma, rotational acceleration of a kinematic element becomes:

$$\dot{\boldsymbol{\omega}} = \underline{\mathbf{x}} \times \ddot{\underline{\mathbf{x}}} + \ddot{\theta}\,\underline{\mathbf{x}} + \dot{\theta}\,\dot{\underline{\mathbf{x}}} \tag{13}$$

Translational Velocity and Acceleration: The translational velocities of a kinematic element are as follows:

$$\dot{\mathbf{B}} = \dot{\mathbf{A}} + \dot{d}\,\underline{\mathbf{x}} + d\,\dot{\underline{\mathbf{x}}}, \qquad \dot{\mathbf{S}} = \dot{\mathbf{A}} + \dot{x}\,\underline{\mathbf{x}} + x\,\dot{\underline{\mathbf{x}}} + \dot{\mathbf{e}} \tag{14}$$

where, thanks to rigidity, the velocity of eccentricity vector is $\dot{\mathbf{e}} = \boldsymbol{\omega} \times \mathbf{e}$. Then, the translational accelerations of a kinematic element are as follows:

$$\ddot{\mathbf{B}} = \ddot{\mathbf{A}} + \ddot{d}\,\underline{\mathbf{x}} + 2\dot{d}\,\dot{\underline{\mathbf{x}}} + d\,\ddot{\underline{\mathbf{x}}}, \qquad \ddot{\mathbf{S}} = \ddot{\mathbf{A}} + \ddot{x}\,\underline{\mathbf{x}} + 2\dot{x}\,\dot{\underline{\mathbf{x}}} + x\,\ddot{\underline{\mathbf{x}}} + \ddot{\mathbf{e}} \tag{15}$$

where the acceleration of eccentricity vector is $\ddot{\mathbf{e}} = \dot{\boldsymbol{\omega}} \times \mathbf{e} + \boldsymbol{\omega} \times \dot{\mathbf{e}}$.

4.2. Dynamics of a Kinematic Element

Active Forces and Torques: Active forces and torques act on a kinematic element. These forces are the distances forces (*e.g.*, gravity) and the actuator forces and torques.

Gravity Forces:. A kinematic element is always subject to gravity. The force of gravity, which is assumed to act at the mass center of a kinematic element, is given as $\mathbf{f_g} = m\,\mathbf{g}$. Here, m is the mass of the kinematic element and $\mathbf{g}$ is the gravity acceleration vector oriented towards the center of the Earth.

Actuator Forces and Torques:. An actuator or actuators can generate mobility for a kinematic element. We write these actuator force and torque vectors $(\mathbf{f}_d,\, \boldsymbol{\tau}_{\underline{\mathbf{x}}},\, \boldsymbol{\tau}_\theta)$, which move a kinematic element, as follows:

$$\mathbf{f}_d = f_d\,\underline{\mathbf{x}}, \qquad \boldsymbol{\tau}_{\underline{\mathbf{x}}} = \tau_{\underline{\mathbf{x}}}\,\underline{\mathbf{z}}, \qquad \boldsymbol{\tau}_\theta = \tau_\theta\,\underline{\mathbf{x}} \tag{16}$$

where f_d, $\tau_{\underline{\mathbf{x}}}$ and τ_θ are scalar force and torques of the actuators, respectively.

Reactive Forces and Torques: Reactive forces and torques appear due to the generated motion: on the one hand, inertia of the kinematic element body in all cases and, optionally, of its associated actuators; on the other hand, friction along the kinematic element (passive or actuated) mobilities.

Body Inertial Forces and Torques:. The reactive linear and angular momentums $(\mathbf{f}^*, \boldsymbol{\tau}^*)$ of a kinematic element body can be calculated with the Newton-Euler equations given in [29],[30].

Actuator Inertial Forces and Torques:. If the extrinsic or intrinsic mobilities are actuated, then one must account for the associated actuator inertias. The rotary actuator inertial torque $\boldsymbol{\tau}_{\underline{\mathbf{x}}}^*$, which appears due to the extrinsic torque $\boldsymbol{\tau}_{\underline{\mathbf{x}}}$, can be written as follows:

$$\boldsymbol{\tau}_{\underline{\mathbf{x}}}^* = -\mathcal{I}_z\left(\ddot{q}\,\underline{\mathbf{z}}\right) = -\mathcal{I}_z\left(\underline{\mathbf{x}} \times \ddot{\underline{\mathbf{x}}}\right) \tag{17}$$

where $\mathcal{I}_z$ is the rotary inertia of the actuator around the $\underline{\mathbf{z}}$ axis and $\ddot{q}$ is the angular acceleration of the actuator. The rotary actuator inertial torque $\boldsymbol{\tau}_{\theta}^*$, which appears due to the intrinsic torque $\boldsymbol{\tau}_{\theta}$, can be written as below:

$$\boldsymbol{\tau}_{\theta}^* = -\mathcal{I}_x\left(\ddot{\theta}\,\underline{\mathbf{x}}\right) \tag{18}$$

where $\mathcal{I}_x$ is the rotary inertia of the actuator around the $\underline{\mathbf{x}}$ axis. Finally, the inertial force of the linear actuator can be written as $\mathbf{f}_d^* = -m_d\,\ddot{d}\,\underline{\mathbf{x}}$, where m_d is the mass moved inside the kinematic element.

Frictional Forces and Torques: Frictional forces and torques $(\bar{\mathbf{f}}_d,\ \bar{\boldsymbol{\tau}}_{\underline{\mathbf{x}}},\ \bar{\boldsymbol{\tau}}_\theta)$ offer resistance on the kinematic element mobility. The extrinsic frictional torque of a rotating kinematic element can be calculated as follows:

$$\bar{\boldsymbol{\tau}}_{\underline{\mathbf{x}}} = -\left(\bar{\tau}_{v(\underline{\mathbf{x}})}\,\dot{q} + \bar{\tau}_{c(\underline{\mathbf{x}})}\,sign(\dot{q})\right)\underline{\mathbf{z}} = -\bar{\tau}_{v(\underline{\mathbf{x}})}\,\boldsymbol{\omega} - \bar{\tau}_{c(\underline{\mathbf{x}})}\,sign(\boldsymbol{\omega}^T\underline{\mathbf{z}})\,\underline{\mathbf{z}} \tag{19}$$

where $\bar{\tau}_{v(\underline{\mathbf{x}})}$ and $\bar{\tau}_{c(\underline{\mathbf{x}})}$ are the viscous and Coulomb friction coefficients of the joint (q) that gives extrinsic mobility of the kinematic element; $\boldsymbol{\omega}$ is the relative rotational velocity vector between the extrinsic joint and the rest of the kinematic element; and $\underline{\mathbf{z}}$ is the axis of rotation of the kinematic element. The intrinsic frictional torque of a self-rotating kinematic element can be calculated as follows:

$$\bar{\boldsymbol{\tau}}_\theta = -\left(\bar{\tau}_{v(\theta)}\,\dot{\theta} + \bar{\tau}_{c(\theta)}\,sign(\dot{\theta})\right)\underline{\mathbf{x}} \tag{20}$$

where $\bar{\tau}_{v(\theta)}$ and $\bar{\tau}_{c(\theta)}$ are the viscous and Coulomb friction coefficients of the joint that gives intrinsic rotational mobility (θ) of the kinematic element whose orientation is along $\underline{\mathbf{x}}$. The intrinsic frictional force of an intrinsically translating kinematic element can be calculated as below:

$$\bar{\mathbf{f}}_d = -\left(\bar{f}_{v(d)}\,\dot{d} + \bar{f}_{c(d)}\,sign(\dot{d})\right)\underline{\mathbf{x}} \tag{21}$$

where $\bar{f}_{v(d)}$ and $\bar{f}_{c(d)}$ are the viscous and Coulomb friction coefficients of the joint that gives intrinsic translational mobility (d) of the kinematic element whose direction is $\underline{\mathbf{x}}$.

5. Kinematic and Dynamic Model of a Parallel Robot

Now, it is time to assemble the kinematic elements into a parallel robot and to write the robot model. To represent the robot motion, both in kinematics and dynamics, let us introduce the following vocabulary and the global modeling scenario:

State Variables [Kane's generalized coordinates] are a chosen set of variables used to define the positions, velocities, accelerations and forces of all the kinematic elements in the mechanism. Note that we do not put any constraint on the *minimality*, which is inherited from serial robots, contrarily we allow for *redundancy* of the set.

Motion Basis [Kane's generalized speeds] is a chosen set of first order derivatives of variables (*e.g.*, scalars, vectors, functions) whose linear combination expresses the motion of the mechanism (*i.e.*, the Cartesian velocities of all the kinematic elements). Normally, a basis in linear algebra is a set of linearly independent vectors in the same space, but here we let it also be a dependent set of non-homogeneous variables for simplicity of the equations.

Motion Constraints [Change of motion basis] map the motion of the mechanism expressed in a redundant motion basis into the minimal motion basis made of the actuator axes.

Kinematic Coordinates [Kane's partial velocities] express the velocity of the mechanism in a given motion basis.

Dynamic Coordinates [Kane's generalized forces] define the dynamic equilibrium of the mechanism in a given motion basis. They are computed from the active and reactive forces (inertia and friction) of all the kinematic elements in the mechanism.

In the light of d'Alembert's principle of virtual work, the equations of motion of the mechanism take the following form [31]:

$$\mathbb{F}^T \mathbf{u} = 0 \qquad (22)$$

where $\mathbb{F} \in \Re^k$ is the vector of dynamic coordinates acting on the redundant motion basis $\mathbf{u} \in \Re^k$ (virtual displacement). This implies that sum of all the exerted efforts on the chosen redundant motion basis should vanish. Equation (22) can be rewritten in terms of the minimal motion basis (*i.e.*, actuation space $\dot{\mathbf{q}} \in \Re^n$, $k > n$) through the motion constraints $M \in \Re^{k \times n}$ as below:

$$\mathbb{F}^T (M \dot{\mathbf{q}}) = 0 \qquad (23)$$

since the above system is defined at minimal motion basis, then it deduces to the following final form:

$$M^T \, \mathbb{F} \; = \; \mathbf{0} \qquad (24)$$

where the dynamic coordinates are calculated from Kane's method [19]. Dynamic coordinates are obtained as follows: (i) Projecting the resultant forces of each of the kinematic elements through the kinematic coordinates onto the axes of the chosen motion basis of the mechanism; (ii) Then, summing separately all the projected contributions on each of the axes of the motion basis gives the dynamic coordinates of the mechanism. This is equivalent to the projection of the external forces onto the joint forces through the transpose jacobian, except that now, the dynamic coordinates are not necessarily the actuator forces and that the kinematic coordinates are not necessarily expressed between end-effector and joint velocities. Here, the kinematic coordinates express rather the velocities of all the kinematic elements, namely the velocity of the whole mechanism.

5.1. State Variables and Motion Basis of a Parallel Robot

5.1.1. State Variables

Instead of writing the geometric relations only with the usual independent number (n_q) of *generalized scalar coordinates* $\{q_1, \ldots, q_{n_q}\}$ (*i.e.*, active joint values), we break out of the customary routine and express them with the unit direction vectors of all (n_{ke}) kinematic elements $\{\underline{\mathbf{x}}_1, \ldots, \underline{\mathbf{x}}_{n_{ke}}\}$, the varying lengths of these kinematic elements $\{d_1, \ldots, d_{n_{ke}}\}$, and the self-rotation angles around their unit directions of these kinematic elements $\{\theta_1, \ldots, \theta_{n_{ke}}\}$, namely with a redundant set of variables $(5 n_{ke} \gg n_q)$.

5.1.2. Motion Basis

Choosing a motion basis, different from the first order derivatives of the independent generalized coordinates $\{\dot{q}_1, \ldots, \dot{q}_{n_q}\}$ of a robot, was proposed for the first time by Kane [19]. He called this motion basis as *generalized speeds*. Traditionally, in Kane's method, the generalized speeds u_r are functions of the first order derivatives of n_q independent scalar coordinates:

$$u_r \; \triangleq \; \sum_{i=1}^{n_q} \mathbf{y}_{ri}(\, q_1, \ldots, q_{n_q} \,) \, \dot{q}_i \; + \; \mathbf{z}_r(\, q_1, \ldots, q_{n_q} \,), \quad r = 1, \ldots, n_q \qquad (25)$$

where $\mathbf{y}_{ri}$ and $\mathbf{z}_r$ are functions of $\{q_1, \ldots, q_{n_q}\}$ and the time t. The choice of these functions in (25) should yield a unique solution for $\{\dot{q}_1, \ldots, \dot{q}_{n_q}\}$ [19].

14

Since the redundant set that we proposed compactly represents the configuration of the mechanism and linearizes the expressions, the choice of the motion basis (*i.e.*, generalized speeds) appears spontaneously itself. So, without needing to inspect the expressions, we define directly the motion basis as the time derivatives of the state variables explained in Section 5.1.1:

$$\mathbf{u}_{xi} \triangleq \dot{\underline{\mathbf{x}}}_i, \quad u_{di} \triangleq \dot{d}_i, \quad u_{\theta i} \triangleq \dot{\theta}_i, \quad i = 1, \ldots, n_{ke} \tag{26}$$

Namely, $\mathbf{y}_{ri} = 1$ and $\mathbf{z}_r = 0$ in (25). *This definition preserves the geometric intuitiveness of the mechanism and eases the following of equations.*

5.2. Kinematics of a Parallel Robot

For the sake of clarity, we will illustrate the method on a simple 2 degrees of freedom (dof) five-bar mechanism which is a RRR-RR structure planar parallel robot. Figure 3 illustrates this 2 dof five-bar mechanism. Regarding defined kinematic element types, a kinematic leg of this robot is composed of two consecutive *bar type* kinematic elements, and this five-bar mechanism can be renamed as a 2BB parallel robot. The revolute joints rotate around the $\underline{\mathbf{z}}_{pi}$ and $\underline{\mathbf{z}}_{ai}$ axes which are orthogonal to the paper plane. Actuators are located at $\mathbf{P}_i$ points. All the kinematic elements are homogenous and symmetric. ℓ_{pi} and ℓ_{ai} are the constant lengths of the kinematic elements. On the right side of the figure, we see the joint-oriented kinematic graph and the new body-oriented kinematic graph of the five-bar mechanism. The five-bar does not have a moving-platform, however one can imagine one of the identical kinematic elements as a pseudo moving platform (*e.g.*, $[\mathbf{A}_2\mathbf{E}]$). The end-effector is located at point $\mathbf{E}$.

5.2.1. Mass Centers

The mass center position of the ith kinematic element of a kinematic leg (with respect to a constant attachment point $\mathbf{P}$ of the kinematic leg onto the base) can be formulated by summing the $i - 1$ elements and adding finally the ith mass center:

$$\mathbf{S}_i = \mathbf{A}_i + x_i \underline{\mathbf{x}}_i + \mathbf{e}_i \tag{27}$$

where

$$\mathbf{A}_i = \mathbf{P} + \sum_{j=1}^{i-1} \overrightarrow{\mathbf{A}_j \mathbf{A}_{j+1}} = \mathbf{P} + \sum_{j=1}^{i-1} d_j \underline{\mathbf{x}}_j \tag{28}$$
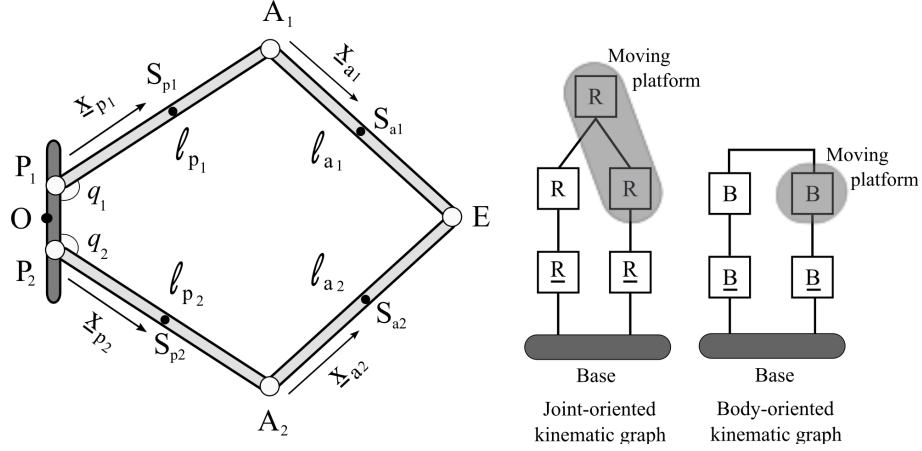
Figure 3: A 2-dof planar five-bar mechanism.

Example:. So, assuming that all the kinematic elements of the five-bar mechanism are homogenous and symmetric, the mass center positions of the kinematic elements shown in Fig. 3 can be simply expressed as follows:

$$\mathbf{S}_{pi} = \mathbf{P}_i + \frac{\ell_{pi}}{2}\,\underline{\mathbf{x}}_{pi}, \qquad \mathbf{S}_{ai} = \mathbf{P}_i + \ell_{pi}\,\underline{\mathbf{x}}_{pi} + \frac{\ell_{ai}}{2}\,\underline{\mathbf{x}}_{ai}, \qquad i = 1,2 \quad (29)$$

where $\mathbf{P}_i$ is a constant point, $\{\underline{\mathbf{x}}_{pi}, \underline{\mathbf{x}}_{ai}\}$ are the unit direction vectors defining the state of the kinematic leg, and $\{\ell_{pi}, \ell_{ai}\}$ are the constant lengths of the kinematic elements.

5.2.2. Velocities

Translational velocity:. The mass center velocity of the ith kinematic element of a kinematic leg can be formulated by simply time differentiating (27), which yields:

$$\dot{\mathbf{S}}_i = \dot{\mathbf{A}}_i + \dot{x}_i\,\underline{\mathbf{x}}_i + x_i\,\underline{\dot{\mathbf{x}}}_i + \dot{\mathbf{e}}_i = \sum_{j=1}^{i-1}\left(\dot{d}_j\,\underline{\mathbf{x}}_j + d_j\,\underline{\dot{\mathbf{x}}}_j\right) + \dot{x}_i\,\underline{\mathbf{x}}_i + x_i\,\underline{\dot{\mathbf{x}}}_i + \dot{\mathbf{e}}_i \quad (30)$$

Example:. So, the velocities of the mass centers $\mathbf{S}_{pi}$ and $\mathbf{S}_{ai}$ of the five-bar mechanism shown in Fig. 3 are written as follows:

$$\dot{\mathbf{S}}_{pi} = \frac{\ell_{pi}}{2}\,\underline{\dot{\mathbf{x}}}_{pi}, \qquad \dot{\mathbf{S}}_{ai} = \ell_{pi}\,\underline{\dot{\mathbf{x}}}_{pi} + \frac{\ell_{ai}}{2}\,\underline{\dot{\mathbf{x}}}_{ai}, \qquad i = 1,2 \quad (31)$$

Rotational velocity:. From *lemma* in Section 4.1, rotational velocity $\boldsymbol{\omega}_i$ of any kinematic element in a kinematic leg, expressed in a fixed reference frame with respect to the same fixed reference frame will be equal to (12).

Example:. So, the rotational velocity vectors of the kinematic elements $\{\mathbf{PA}\}_i$ and $\{\mathbf{AE}\}_i$ of the ith kinematic leg of the five-bar mechanism given in Fig. 3 can be calculated as below:

$$\boldsymbol{\omega}_{pi} \triangleq \underline{\mathbf{x}}_{pi} \times \dot{\underline{\mathbf{x}}}_{pi}, \qquad \boldsymbol{\omega}_{ai} \triangleq \underline{\mathbf{x}}_{ai} \times \dot{\underline{\mathbf{x}}}_{ai}, \qquad i = 1, 2 \qquad (32)$$

Since the rotations around the orientation vectors $\underline{\mathbf{x}}_i$ of the kinematic elements are not allowed, the terms $\dot{\theta}_i \underline{\mathbf{x}}_i$ drop.

5.2.3. Accelerations

Translational acceleration:. The mass center acceleration of the ith kinematic element of a kinematic leg is as below:

$$\ddot{\mathbf{S}}_i = \sum_{j=1}^{i-1} \left(\ddot{d}_j \underline{\mathbf{x}}_j + 2\,\dot{d}_j \dot{\underline{\mathbf{x}}}_j + d_j \ddot{\underline{\mathbf{x}}}_j \right) + \ddot{x}_i \underline{\mathbf{x}}_i + 2\dot{x}_i \dot{\underline{\mathbf{x}}}_i + x_i \ddot{\underline{\mathbf{x}}}_i + \ddot{\mathbf{e}}_i \quad (33)$$

Rotational acceleration:. Rotational acceleration vector $\dot{\boldsymbol{\omega}}_i$ of the ith (with respect to base) kinematic element of a kinematic leg will be equal to (13).

5.3. Kinematic Constraints of a Parallel Robot

5.3.1. Configuration Constraints

If the positions, orientations, and lengths of the kinematic elements of a robot are restricted by the presence of each other's contacts, then the robot is said to be subject to *configuration constraints.* Such restrictions are expressed through the implicit kinematic model (ImplKM) of the robot or so-called the *holonomic constraint equation* [19]:

$$f\left(\mathbf{O},\, \mathbf{P}_i,\, \mathbf{E},\, \left\{ \underline{\mathbf{x}}_{ji},\, d_{ji},\, (\theta_{ji}) \right\}_{j=1,\dots,n_{ke(i)}},\, \boldsymbol{\xi}_{geo} \right) = \mathbf{0}, \quad \forall i = 1,\dots, n_{leg} \quad (34)$$

where $\boldsymbol{\xi}_{geo}$ is the vector of constant geometric parameters, n_{leg} is the number of kinematic legs, and $n_{ke(i)}$ is the number of kinematic elements in the ith kinematic leg of a parallel robot. Assuming that: the connection points of the kinematic elements are lying on the axes of the direction vectors; the self-rotations of the kinematic elements θ_{ji} do not change the positions of these connection points; and the end-effector frame is located at the mass center of the moving platform; then equation (34) can be precisely rewritten as follows:

$$\overrightarrow{\mathbf{OE}} - \sum_{j=1}^{n_{ke(i)}} d_{ji}\, \underline{\mathbf{x}}_{ji} - \overrightarrow{\mathbf{OP}}_i = \mathbf{0}, \qquad i = 1,\dots, n_{leg} \qquad (35)$$

where the sum $n_{ke(1)} + \dots + n_{ke(n_{leg})} = n_{ke}$ is equal to the total number of kinematic elements in a parallel robot.

Example:. For the five-bar shown in Fig. 3, the *closed-loop holonomic constraint equations* of the kinematic elements can be written as follows:

$$\overrightarrow{\mathbf{OE}} - \ell_{ai}\,\underline{\mathbf{x}}_{ai} - \ell_{pi}\,\underline{\mathbf{x}}_{pi} - \overrightarrow{\mathbf{OP}}_i = \mathbf{0}\,, \qquad i = 1,2 \tag{36}$$

5.3.2. Motion Constraints

If the components of motion basis $\{\,\dot{\underline{\mathbf{x}}}_i,\ \dot{d}_i,\ \dot{\theta}_i\,\}$ of the mechanism are not mutually independent, then the mechanism is said to be subject to *motion constraints*, and the mechanism is named as a *nonholonomic system*. The motion constraints equation can be written by differentiating the configuration constraints equation (35), which gives:

$$M_{Ci}\,\dot{\mathbb{X}} - \sum_{j=1}^{n_{ke(i)}} \left(\dot{d}_{ji}\,\underline{\mathbf{x}}_{ji} + d_{ji}\,\dot{\underline{\mathbf{x}}}_{ji}\right) = \mathbf{0}\,, \qquad i = 1,\ldots,n_{leg} \tag{37}$$

where $\mathbb{X}$ is a Cartesian pose representation of the end-effector in a column-array form, and where $M_{Ci} \in \Re^{3 \times dim(\mathbb{X})}$ is the Cartesian position kinematic matrix which relates the velocity of the Cartesian pose to the velocity of the tip position of the ith kinematic leg. M_{Ci} depends on the representation of $\mathbb{X}$, the structure of the rigid moving platform (or the articulated nacelle), and the tip position of the ith kinematic leg on the moving platform (or nacelle).

Example:. For the five-bar shown in Fig. 3, the motion constraint equations are written by differentiating (36) as follows:

$$\dot{\mathbb{X}} - \ell_{pi}\,\dot{\underline{\mathbf{x}}}_{pi} - \ell_{ai}\,\dot{\underline{\mathbf{x}}}_{ai} = \mathbf{0}\,, \qquad i = 1,2 \tag{38}$$

where $\dot{\mathbb{X}} = \dot{\mathbf{E}}$. From (38), we can derive the inverse differential kinematic models of the kinematic elements' variables. To do so, we exploit two properties of the vectors:

- Projection of a vector onto its velocity vector is equal to zero: $\underline{\mathbf{x}}^T \dot{\underline{\mathbf{x}}} = 0$,
- If vectors $\mathbf{a}$ and $\mathbf{c}$ are parallel ($\mathbf{a}//\mathbf{c}$), then $\mathbf{a}(\mathbf{b}^T\mathbf{c}) = (\mathbf{b}^T\mathbf{a})\mathbf{c}$.

By projecting $\underline{\mathbf{x}}_{ai}$ onto (38), we eliminate its velocity $\dot{\underline{\mathbf{x}}}_{ai}$ from the equation:

$$\underline{\mathbf{x}}_{ai}^T \dot{\mathbb{X}} - \ell_{pi}\,\underline{\mathbf{x}}_{ai}^T \dot{\underline{\mathbf{x}}}_{pi} = 0 \tag{39}$$

Afterwards, multiplying (39) with $\underline{\mathbf{y}}_{pi}$, which is parallel to $\dot{\underline{\mathbf{x}}}_{pi}$, and then using the second property of the vectors mentioned above, we rewrite (39) as follows:

$$\underline{\mathbf{y}}_{pi}\,\underline{\mathbf{x}}_{ai}^T \dot{\mathbb{X}} - \ell_{pi}\,(\underline{\mathbf{x}}_{ai}^T \underline{\mathbf{y}}_{pi})\,\dot{\underline{\mathbf{x}}}_{pi} = \mathbf{0} \tag{40}$$

18

This avoids a matrix inversion while computing the inverse differential kinematic model of $\dot{\underline{\mathbf{x}}}_{pi}$:

$$\dot{\underline{\mathbf{x}}}_{pi} \;=\; M_{pi}\,\dot{\mathbb{X}}, \qquad M_{pi} \;=\; \left[\frac{\underline{\mathbf{y}}_{pi}\,\mathbf{x}_{ai}^{T}}{\ell_{pi}\,(\,\mathbf{x}_{ai}^{T}\,\underline{\mathbf{y}}_{pi}\,)}\right] \in \Re^{3\times 3} \tag{41}$$

Then, to derive the other inverse differential kinematic model related to $\dot{\underline{\mathbf{x}}}_{ai}$, we proceed as follows:

$$\dot{\mathbb{X}} \;-\; \ell_{pi}\,M_{pi}\,\dot{\mathbb{X}} \;-\; \ell_{ai}\,\dot{\underline{\mathbf{x}}}_{ai} \;=\; \mathbf{0} \tag{42}$$

and from (42) we write easily:

$$\dot{\underline{\mathbf{x}}}_{ai} \;=\; M_{ai}\,\dot{\mathbb{X}}, \qquad M_{ai} \;=\; \left[\frac{1}{\ell_{ai}}\,(\,I_3\,-\,\ell_{pi}\,M_{pi}\,)\right] \in \Re^{3\times 3} \tag{43}$$

where I_3 is the 3 by 3 identity matrix. Finally, we derive the inverse differential kinematic model of the active joint coordinates $\dot{q}_i$. Knowing that:

$$\boldsymbol{\omega}_{pi} \;\triangleq\; \underline{\mathbf{x}}_{pi} \,\times\, \dot{\underline{\mathbf{x}}}_{pi} \;=\; \dot{q}_i\,\mathbf{z}_{pi} \tag{44}$$

we can take out $\dot{q}_i$ as below:

$$\dot{q}_i \;=\; (\,\underline{\mathbf{x}}_{pi} \,\times\, \dot{\underline{\mathbf{x}}}_{pi}\,)^{T}\,\mathbf{z}_{pi} \tag{45}$$

which can be reformulated in terms of $\dot{\mathbb{X}}$ as follows:

$$\dot{q}_i \;=\; M_{qi}\,\dot{\mathbb{X}}, \qquad M_{qi} \;=\; \left[\,\mathbf{z}_{pi}^{T}\,[\underline{\mathbf{x}}_{pi}]_{\times}\,M_{pi}\,\right] \in \Re^{1\times 3} \tag{46}$$

where $[\cdot]_{\times}$ represents the skew-symmetric matrix of an associated cross-product vector.

5.4. Kinematic Coordinates of a Parallel Robot

Kane [19] expresses the linear and rotational velocities of kinematic elements with the independent generalized scalar coordinates (*i.e.*, state variables), the generalized speeds (*i.e.*, a motion basis) defined in (25) and the *partial velocities* (*i.e.*, kinematic coordinates). Then, Kane writes the linear velocity for the mass center and the rotational velocity of a kinematic element as follows:

$$\dot{\mathbf{S}} \;=\; \sum_{r=1}^{n_q} \mathbf{v}_r\,u_r \;+\; \mathbf{v}_t\,, \qquad \boldsymbol{\omega} \;=\; \sum_{r=1}^{n_q} \mathbf{w}_r u_r \;+\; \mathbf{w}_t \tag{47}$$

where $\mathbf{v}_r$, $\mathbf{w}_r$, $\mathbf{v}_t$ and $\mathbf{w}_t$ are functions of $\{q_1, \ldots, q_{n_q}\}$ and the time t. The vectors $\mathbf{v}_r \in \Re^3$ and $\mathbf{w}_r \in \Re^3$ are the rth *partial linear and rotational velocities* of the kinematic element. So, for a kinematic element, Kane defines n_q partial linear velocities and n_q partial rotational velocities with the use of n_q scalar generalized speeds:

$$\mathbf{v}_r = \partial \dot{\mathbf{S}} / \partial u_r, \qquad \mathbf{w}_r = \partial \boldsymbol{\omega} / \partial u_r, \qquad r = 1, \ldots, n_q \qquad (48)$$

Regarding the definition of our motion basis (direction vectors, lengths and rotation angles) in (26), the kinematic coordinates take the form of either matrices or vectors:

$$V_{xi} = \partial \dot{\mathbf{S}} / \partial \mathbf{u}_{xi}, \qquad \mathbf{v}_{di} = \partial \dot{\mathbf{S}} / \partial u_{di}, \qquad \mathbf{v}_{\theta i} = \partial \dot{\mathbf{S}} / \partial u_{\theta i} \qquad (49)$$

$$W_{xi} = \partial \boldsymbol{\omega} / \partial \mathbf{u}_{xi}, \qquad \mathbf{w}_{di} = \partial \boldsymbol{\omega} / \partial u_{di}, \qquad \mathbf{w}_{\theta i} = \partial \boldsymbol{\omega} / \partial u_{\theta i} \qquad (50)$$

where $V_{xi} \in \Re^{3 \times 3}$ and $W_{xi} \in \Re^{3 \times 3}$ are the linear and rotational kinematic coordinates (matrices) of the kinematic element with respect to the ith kinematic element's direction vector variable $\underline{\mathbf{x}}_i$; and where $\mathbf{v}_{di} \in \Re^3$ and $\mathbf{w}_{di} \in \Re^3$ are the linear and rotational kinematic coordinates (vectors) of the kinematic element with respect to the ith kinematic element's length variable d_i; and where $\mathbf{v}_{\theta i} \in \Re^3$ and $\mathbf{w}_{\theta i} \in \Re^3$ are the linear and rotational kinematic coordinates (vectors) of the kinematic element with respect to the ith kinematic element's self-rotation variable θ_i; and where $i = 1, \ldots, n_{ke}$.

Example:. For the five-bar shown in Fig. 3, the motion basis will be as follows:

$$\mathbf{u}_{x1} \triangleq \dot{\underline{\mathbf{x}}}_{p1}, \qquad \mathbf{u}_{x2} \triangleq \dot{\underline{\mathbf{x}}}_{p2} \qquad \mathbf{u}_{x3} \triangleq \dot{\underline{\mathbf{x}}}_{a1} \qquad \mathbf{u}_{x4} \triangleq \dot{\underline{\mathbf{x}}}_{a2} \qquad (51)$$

Table 1 tabulates the linear and rotational kinematic coordinates of this five-bar mechanism. These kinematic coordinates are algebraic expressions written from the geometric states of the kinematic elements.

Table 1: The (transposed) kinematic coordinates of the five-bar mechanism, i=1,2.

	$\partial \dot{\mathbf{S}}_{pi}$	$\partial \boldsymbol{\omega}_{pi}$	$\partial \dot{\mathbf{S}}_{ai}$	$\partial \boldsymbol{\omega}_{ai}$
$\partial \dot{\underline{\mathbf{x}}}_{pi}$	$\frac{\ell_{pi}}{2} I_3$	$[\underline{\mathbf{x}}_{pi}]_\times^T$	$\ell_{pi} I_3$	$\mathbf{0}$
$\partial \dot{\underline{\mathbf{x}}}_{ai}$	$\mathbf{0}$	$\mathbf{0}$	$\frac{\ell_{ai}}{2} I_3$	$[\underline{\mathbf{x}}_{ai}]_\times^T$

Remark: We also remark that while calculating the kinematic coordinates of a kinematic leg, the kinematic leg will have contributions only from itself since it can solely be represented by its own state variables. The contributions from the rest of kinematic legs will be zero.

5.5. Dynamic Coordinates of a Parallel Robot

Once the exhaustive list of forces and torques acting on each kinematic element in the parallel robot is complete (see Table 2 for the case of the five-bar mechanism), then one needs to distinguish between *contributing* and *non-contributing* forces and torques in the mechanism. Indeed, the dynamic model of the robot only depends on the *contributing* ones which yield the dynamic coordinates (*i.e.*, generalized forces).

Table 2: The local forces and torques of the five-bar mechanism, i=1,2.

	Active		*Friction*		*Inertia*	
	Actuator	*Gravity*	*Actuator*	*PassiveJoint*	*Actuator*	*Element*
$Forces(pi)$	$\mathbf{0}$	$\mathbf{f_{g}}_{(pi)}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{f}^*_{pi}$
$Torques(pi)$	$\tau_{\underline{\mathbf{x}}_{pi}}\underline{\mathbf{z}}_{pi}$	$\mathbf{0}$	$\bar{\tau}_{\underline{\mathbf{x}}_{pi}}$	$\mathbf{0}$	$\boldsymbol{\tau}^*_{\underline{\mathbf{x}}_{pi}}$	$\boldsymbol{\tau}^*_{pi}$
$Forces(ai)$	$\mathbf{0}$	$\mathbf{f_{g}}_{(ai)}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{f}^*_{ai}$
$Torques(ai)$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\bar{\tau}_{\underline{\mathbf{x}}_{ai}}$	$\mathbf{0}$	$\boldsymbol{\tau}^*_{ai}$

Formula of dynamic coordinates, which is explained in Kane's method, is here rewritten by taking into account the previous *remark*. In dynamic coordinates, the *non-contributing forces* are eliminated by projecting the resultant forces and torques of the kinematic elements onto their motion directions:

$$(\mathbb{F})_u = \sum_{j=1}^{n_{ke(i)}} \left(\left[\left(\frac{\partial \dot{\mathbf{S}}_j}{\partial u} \right)^T \ \left(\frac{\partial \boldsymbol{\omega}_j}{\partial u} \right)^T \right] \begin{bmatrix} \mathbf{f}_{R_j} \\ \boldsymbol{\tau}_{R_j} \end{bmatrix} \right) \tag{52}$$

$$i \in \{1, \ldots, n_{leg}\}, \qquad u \in \{\dot{\underline{\mathbf{x}}}_{bi}, \dot{d}_{bi}, \dot{\theta}_{bi}\}, \qquad b \in \{1, \ldots, n_{ke(i)}\}$$

where $n_{ke(i)}$ is the number of kinematic elements in kinematic leg i of a parallel robot, $\mathbf{f}_{R_j}$ and $\boldsymbol{\tau}_{R_j}$ are the resultant force and torque equal to the sum of all the forces and torques acting on kinematic element j. Here u indicates to which axis of the motion basis that a computed dynamic coordinate corresponds. For each axis of the motion basis, one dynamic coordinate is computed. Each kinematic element has 3 dynamic coordinates corresponding to its own motion basis axes $\{\dot{\underline{\mathbf{x}}}, \dot{d}, \dot{\theta}\}$. Totally, $3n_{ke}$ dynamic coordinates are computed for n_{ke} kinematic elements.

Example:. The dynamic coordinates of the five-bar mechanism shown in Fig. 3 can be simply computed through the matrix-wise multiplication of

the Tables 1 and 2. In other words, by multiplying the sum of the local forces and torques with the transposed kinematic coordinates:

$$\begin{bmatrix} \mathbb{F}_{\underline{x}_{pi}} \\ \mathbb{F}_{\underline{x}_{ai}} \end{bmatrix} = \begin{bmatrix} \frac{\ell_{pi}}{2}I_3 & [\mathbf{x}_{pi}]_\times^T & \ell_{pi}I_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\ell_{ai}}{2}I_3 & [\mathbf{x}_{ai}]_\times^T \end{bmatrix} \begin{bmatrix} \mathbf{f}_{\mathbf{g}(pi)} + \mathbf{f}_{pi}^* \\ \tau_{\mathbf{x}_{pi}}\underline{z}_{pi} + \bar{\tau}_{\underline{\mathbf{x}}_{pi}} + \tau_{\underline{\mathbf{x}}_{pi}}^* + \tau_{pi}^* \\ \mathbf{f}_{\mathbf{g}(ai)} + \mathbf{f}_{ai}^* \\ \bar{\tau}_{\underline{\mathbf{x}}_{ai}} + \tau_{ai}^* \end{bmatrix} \tag{53}$$

where the dynamic coordinates, $\mathbb{F}_{\underline{x}_{pi}}$ and $\mathbb{F}_{\underline{x}_{ai}}$, are kind of exiting forces effecting the rotations of the bars of the mechanism. In other words, these rotations of the bars are the result of the total work done by these exiting forces along the displacement directions $\dot{\underline{\mathbf{x}}}_{pi}$ and $\dot{\underline{\mathbf{x}}}_{ai}$.

5.6. Dynamic Constraints of a Parallel Robot

Dynamic constraints of a parallel robot can be written from d'Alembert's principle of virtual work as follows:

$$\sum_{i=1}^{n_{ke}} \left(\mathbb{F}_{\underline{x}_i}^T \dot{\underline{\mathbf{x}}}_i + \mathbb{F}_{d_i} \dot{d}_i + \mathbb{F}_{\theta_i} \dot{\theta}_i \right) = 0 \tag{54}$$

where $\mathbb{F}_{\underline{x}_i} \in \Re^3$, $\mathbb{F}_{d_i} \in \Re$ and $\mathbb{F}_{\theta_i} \in \Re$ are the corresponding dynamic coordinates. Dynamic constraints (54) can be reformulated through the known motion constraint models which relate kinematic elements' motions to the velocity of the end-effector pose (to a motion basis of the constraint space):

$$\dot{\underline{\mathbf{x}}}_i = M_{\underline{x}_i} \dot{\mathbb{X}}, \qquad \dot{d}_i = M_{d_i} \dot{\mathbb{X}}, \qquad \dot{\theta}_i = M_{\theta_i} \dot{\mathbb{X}} \tag{55}$$

The substitution of (55) into (54) and subsequently elimination of $\dot{\mathbb{X}}$ yield the following dynamic constraints:

$$\sum_{i=1}^{n_{ke}} \left(M_{\underline{x}_i}^T \mathbb{F}_{\underline{x}_i} + M_{d_i}^T \mathbb{F}_{d_i} + M_{\theta_i}^T \mathbb{F}_{\theta_i} \right) = \mathbf{0} \tag{56}$$

Example:. Exploiting (56), the dynamic constraints of the five-bar mechanism are written as follows:

$$\sum_{i=1}^{2} \left(M_{pi}^T \mathbb{F}_{\underline{x}_{pi}} + M_{ai}^T \mathbb{F}_{\underline{x}_{ai}} \right) = M_p^T \mathbb{F}_{\underline{x}_p} + M_a^T \mathbb{F}_{\underline{x}_a} = \mathbf{0} \tag{57}$$

22

where $\mathbb{F}_{\underline{x}_p} \in \Re^6$ and $\mathbb{F}_{\underline{x}_a} \in \Re^6$ are the stacked vectors of the dynamic coordinates of $\mathbb{F}_{\underline{x}_{pi}} \in \Re^3$ and $\mathbb{F}_{\underline{x}_{ai}} \in \Re^3$, respectively. $M_p \in \Re^{6 \times 3}$ and $M_a \in \Re^{6 \times 3}$ are also stacked matrices of the motion constraint models $M_{pi} \in \Re^{3 \times 3}$ and $M_{ai} \in \Re^{3 \times 3}$, respectively.

5.7. Linear Solution for the Inverse Dynamics

Every equation from the beginning up to the last equation (56) is expressed in a linear form. Therefore, progressing from (56) to a unique linear implicit dynamic model (LImplDM) of a parallel robot is just a matter of some simple linear algebraic manipulations, once the motorized joints are specified:

$$A\,\mathbf{\Gamma} + \mathbf{b} = \mathbf{0} \tag{58}$$

where matrix $A \in \Re^{r \times k}$ is dependant to the mechanism configuration and it relates the unknown force vector $\mathbf{\Gamma} \in \Re^k$ of the actuators to the contributing efforts $\mathbf{b} \in \Re^r$ of the kinematic elements of the mechanism. Here r is the dimension of a surjective motion basis ($r \geq k$). Required parameters and variables to write this LimplDM are as follows:

- ξ_{geo} constant geometric parameters of the robot (*e.g.*, lengths, points).
- ξ_{dyn} constant dynamic parameters of the robot (*e.g.*, masses, inertias).
- $\{\underline{\mathbf{x}}, d, \theta\}$: 0th order variables of the kinematic elements. They allow us to write the static configuration of the robot, the motion constraint models and the kinematic coordinates.
- $\{\dot{\underline{\mathbf{x}}}, \dot{d}, \dot{\theta}\}$, $\{\ddot{\underline{\mathbf{x}}}, \ddot{d}, \ddot{\theta}\}$: 1st and 2nd order variables of the kinematic elements. They allow us to write the local forces and torques.
- $\mathbf{\Gamma}$: the sought-after force vector of the robot's actuators (*e.g.*, forces of prismatic actuators and torques of revolute actuators).

Corollary. *If the configuration-dependent matrix A is full rank, we can then solve for the inverse dynamics (IDM) of a parallel robot:*

$$\mathbf{\Gamma} = -A^{\dagger}\,\mathbf{b} \tag{59}$$

where $A^{\dagger}$ is the pseudo-inverse of the matrix A to be computed numerically with a QR or SVD decomposition for a fast and robust solution rather than literally with the Moore-Penrose formula.

Example:. Here, we solve for the inverse dynamics of the five-bar mechanism shown in Fig. 3. To do so, we first write explicitly the equation of the dynamic coordinate (the first one in (53)) which includes the motor torques:

$$\mathbb{F}_{\underline{x}_{pi}} = [\mathbf{x}_{pi}]_\times^T \mathbf{z}_{pi} \, \tau_{\mathbf{x}_{pi}} + \widetilde{\mathbb{F}}_{\underline{x}_{pi}} = \tau_{\mathbf{x}_{pi}} \mathbf{y}_{\underline{p}i} + \widetilde{\mathbb{F}}_{\underline{x}_{pi}} \tag{60}$$

where

$$\widetilde{\mathbb{F}}_{\underline{x}_{pi}} = \tfrac{\ell_{pi}}{2} \left(\mathbf{f}_{\mathbf{g}(pi)} + \mathbf{f}_{pi}^* \right) + [\mathbf{x}_{pi}]_\times^T \left(\bar{\boldsymbol{\tau}}_{\mathbf{x}_{pi}} + \boldsymbol{\tau}_{\mathbf{x}_{pi}}^* + \boldsymbol{\tau}_{pi}^* \right) + \ell_{pi} \left(\mathbf{f}_{\mathbf{g}(ai)} + \mathbf{f}_{ai}^* \right) \tag{61}$$

Afterwards, we can rewrite the dynamic constraints (57) of the five-bar mechanism as below:

$$M_p^T \left(\begin{bmatrix} \mathbf{y}_{\underline{p}1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{y}_{\underline{p}2} \end{bmatrix} \begin{bmatrix} \tau_{\mathbf{x}_{p1}} \\ \tau_{\mathbf{x}_{p2}} \end{bmatrix} + \begin{bmatrix} \widetilde{\mathbb{F}}_{\underline{x}_{p1}} \\ \widetilde{\mathbb{F}}_{\underline{x}_{p2}} \end{bmatrix} \right) + M_a^T \mathbb{F}_{\underline{x}_a} = \mathbf{0} \tag{62}$$

which can be reformulated in the form of (58):

$$A \begin{bmatrix} \tau_{\mathbf{x}_{p1}} \\ \tau_{\mathbf{x}_{p2}} \end{bmatrix} + \mathbf{b} = \mathbf{0} \tag{63}$$

where $A \in \Re^{3\times2}$ and $\mathbf{b} \in \Re^3$ are as follows:

$$A = M_p^T \begin{bmatrix} \mathbf{y}_{\underline{p}1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{y}_{\underline{p}2} \end{bmatrix}, \qquad \mathbf{b} = M_p^T \begin{bmatrix} \widetilde{\mathbb{F}}_{\underline{x}_{p1}} \\ \widetilde{\mathbb{F}}_{\underline{x}_{p2}} \end{bmatrix} + M_a^T \mathbb{F}_{\underline{x}_a} \tag{64}$$

Finally, the solution $\boldsymbol{\Gamma} = [\tau_{\mathbf{x}_{p1}}, \tau_{\mathbf{x}_{p2}}]^T$ (torque vector of the motors) of inverse dynamics of the five-bar mechanism is calculated as follows:

$$\begin{bmatrix} \tau_{\mathbf{x}_{p1}} \\ \tau_{\mathbf{x}_{p2}} \end{bmatrix} = -A^\dagger \mathbf{b} \tag{65}$$

The computational complexity up to the solution (65) of the five-bar mechanism costs 404 addition (+) and 656 multiplication (*) arithmetic operations.

6. A Global View to the Proposed Method

In short, using the proposed method, one can write *efficiently* the inverse dynamic model of a parallel robot by simply following these 6 steps:

1. Decompose the parallel robot to its kinematic elements (by inspection);
2. Define the type of each of the kinematic elements (by inspection);
3. Compute the kinematic coordinates and constraints (automatic);
4. List the local forces and torques on the kinematic elements (automatic);
5. Compute the dynamic coordinates and constraints (automatic);
6. Solve linearly for the inverse dynamic model (automatic).

The state variables $\{\underline{\mathbf{x}}, d, \theta\}$ can be measured directly with proprioceptive and/or exteroceptive sensors (*e.g.*, motor encoders, camera), and/or be obtained through some kinematic models, if this does not lower the efficiency. Therefore, we generalize the representation of the inverse dynamic model without concern for sensors and the models used as follows:

$$\mathbf{\Gamma} = IDM(\ddot{\mathbf{s}}, \dot{\mathbf{s}}, \mathbf{s}, \xi_{geo}, \xi_{dyn}) = -A^{\dagger}(\mathbf{s})\,\mathbf{b}(\ddot{\mathbf{s}}, \dot{\mathbf{s}}, \mathbf{s}) \qquad (66)$$

where $\mathbf{s}$ is the set of state variables of the kinematic elements:

$$\mathbf{s} : \{\,\underline{\mathbf{x}}_i, d_i, \theta_i\,\}, \quad i = 1, \ldots, n_{ke} \qquad (67)$$

Now, we can remark that this paper gave an answer to the question of finding an efficient set of dynamic state variables.

If this proposed method is compared with Khalil's, Kane's and Tsai's methods, then the following differences can be listed:

- We were inspired by the idea of using passive joint coordinates with the active ones (redundancy) in modeling from Khalil, and we recommended a new redundant set of state variables which keeps the equations compact and linear. Furthermore, in this way we do not need to compute the global balancing force at the end-effector.

- We were inspired by the idea of to be free in our choice of a motion basis (minimal or redundant) from Kane, and we proposed for the first time a unique redundant motion basis which makes Kane's method easily applicable to broad range of robots (serial and parallel).

- We were inspired by the idea of writing final equations of motion easily using the local efforts done on each of the kinematic elements from Tsai, and we improved Tsai's formulation by Khalil and Kane's inspirational ideas such that it became geometrically more intuitive, simpler, completely linear and more practical.

7. Computational Complexity

From a practical point of view, vision allows us to sense the kinematic elements (*i.e.*, directed 3D lines) in Cartesian space, which uniquely define the state of the robot [13], [14]. In [28], we show how the states of the kinematic elements and their velocities ($\underline{\dot{\mathbf{x}}}$, $\dot{d}$, $\dot{\theta}$, $\underline{\mathbf{x}}$, d, θ) can be estimated rapidly with an off-the-self CMOS camera through sequential ROI (region of interest) acquisitions, and as well as we present some simulation results for the dynamic control of the Quattro parallel robot whose inverse dynamic model is derived with the proposed approach and is fed back with a vision-based state observer. Finally, as far as physical implementation is concerned, there now exist high-speed cameras that take 500 4M-pixel images per second with real-time access to the images. Thus, vision has potential to allow for a direct use of our method on the real robots.

Moreover, Table 3 compares the required number of arithmetic operations to calculate the inverse dynamics of the five-bar mechanism presented in this paper assuming that feedbacks are provided either by vision or by encoders ($\dot{\mathbf{q}}$, $\mathbf{q}$). We can see that in Table 3 trying to express dependent variables by means of independent ones increases the computational complexity and introduces non-linear operations. This difference in computational complexity will be pronounced severely for a more complex parallel robot.

Table 3: Computational complexity for the five-bar mechanism.

Operation	Encoders − based	Vision − based
Addition	484	404
Multiplication	778	656
Sine	2	0
Cosine	2	0

8. Conclusions

This paper outlined a linear dynamic modeling method for parallel robots based on observable kinematic elements. This modeling method is easily applied to wide range of parallel robots. In [28], we applied it to the Quattro, Gough-Stewart, Orthoglide, Delta and planar 3RRR parallel robots. Even for such complex robots, the equations are simple and compact. Neither trigonometric nor exponential functions are used. Written equations use only

addition (+) and multiplication (*) operators. There is only the piecewise-linear *signum* function in the Coulomb friction which needs a simple sign check for its implementation.

We remark that, this modeling method is linear on the condition that state variables { $\underline{\mathbf{x}}$, d, θ } of all the kinematic elements in the mechanism and their velocities are given. In [28], we show how to calculate rapidly these state variables using vision so that the proposed method can be integrated for dynamic control of parallel robots.

As a future work: (i) we shall analyse the singularities of the configuration-dependent matrix A of linear implicit dynamic model; and (ii) we shall improve this method for flexible kinematic elements.

Acknowledgment

References

[1] A. Vivas, P. Poignet, and F. Pierrot. Predictive functional control for a parallel robot. *IEEE International Conference on Intelligent Robots and Systems (IROS'03)*, pages 2785–2790, October 2003.

[2] V. Nabat, S. Krut, O. Company, P. Poignet, and F. Pierrot. On the design of a fast parallel robot based on its dynamic model. *Springer-Verlag Berlin Heidelberg, Experimental Robotics*, January 2008.

[3] Alon Wolf and Moshe Shoham. Screw theory for the synthesis of the geometry of a parallel robot for a given instantaneous task. *Mechanism and Machine Theory*, pages 656–670, 2006.

[4] Roy Featherstone. On the limits to invariance in the twist/wrench and motor representations of motion and force vectors. *Proc. of A Symp. Commemorating the Legacy, Works, and Life of Sir R.S. Ball. Upon the 100th Anniversary of* A Treatise on the Theory of Screws, 2000.

[5] R.M. Murray, Z. Li, and S.S. Sastry. A mathematical introduction to robotic manipulation. *CRC Press*, 1994.

[6] H. Bruyninckx and J. De Schutter. Unified kinetostatics for serial, parallel and mobile robots. *Advances in Robotic Kinematics (ARK'98)*, pages 343–352, 1998.

[7] L. Riberio, R. Guenther, and D. Martins. Screw-based relative Jacobian for manipulators cooperating in a task. *ABCM Symposium Series in Mechatronics*, pages 276–285, 2008.

[8] L.W. Tsai. The Jacobian analysis of a parallel maniuplator using the theory of reciprocal screws. *Thechnical Research Report*, 1998.

[9] M.G. Mohammed and J. Duffy. A direct analysis of instantaneous kinematics of fully parallel robot manipulators. *ASME Journal of Mechanism, Transmissions, and Automation in Design*, pages 226–229, 1985.

[10] L.W. Tsai. Robot analysis: The mechanics of serial and parallel manipulators. *John Wiley & Sons, Inc.*, 1999.

[11] E. Staffetti and F. Thomas. Analytic formulation of the kinestatics of robot manipulators with arbitrary topology. *IEEE Int. Conf. on Robotics and Automation, (ICRA'02)*, 2002.

[12] L. Baron and J. Angeles. The on-line direct kinematics of parallel manipulators under joint sensor redundancy. *Advances in Robot Kinematics (ARK'98)*, 1998.

[13] N. Andreff and P. Martinet. Vision-based kinematic modelling of some parallel manipulators for control purposes. *In 1st European Conference on Mechanism Science (EuCoMeS'06)*, 2006.

[14] N. Andreff, T. Dallej, and P. Martinet. Image-based visual servoing of a Gough-Stewart parallel manipulator using leg observations. *International Journal of Robotics Research. Special Issue on Vision and Robotics - Joint with the Int. Journal on Computer Vision*, 2007.

[15] P. Mitiguy and T. Kane. Motion variables leading to efficient equations of motion. *Journal of Robotics Research*, pages 522–532, October 1996.

[16] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. *IEEE Int. Conf. on Robotics and Automation, (ICRA'04)*, New Orleans, LA, 2004.

[17] M.J. Liu, C.X. Li, and C.N. Li. Dynamics analysis of the Gough Stewart platform manipulator. *IEEE Trans. on Robotics and Automation*, 16(1), 2000.

[18] Q. Meng, T. Zhang, J.F. He, J.Y. Song, and J.W. Han. Dynamic modeling of a 6-degreeof-freedom stewart platform driven by a permanent magnet synchronous motor. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)*, 11(10), 2010.

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

[20] F. Paccot, N. Andreff, and P. Martinet. A review on dynamic control of parallel kinematic machines: theory and experiments. *Int. Journal of Robotics Research*, pages 395–416, February 2009.

[21] K.J. Waldron and K.H. Hunt. Series-parallel dualities in actively coordinated mechanisms. *Int. J. of Robotics Research*, pages 473–480, 1991.

[22] H. Bruyninckx. Dualities between serial and parallel 321 manipulators. *In International Conference on Robotics and Automation (ICRA'99)*, pages 1532–1537, Detroit, Michigan, May 1999.

[23] T. Dallej. Contribution à un modèle générique pour l'asservissement visuel des robots parallèles par observation des éléments cinématiques. *Thèse de Doctorat, Université Blaise Pascal*, 2007.

[24] T.G. Ionescu. Terminology for mechanisms and machine science. *Mechanism and Machine Theory*, pages 597–901, 2003.

[25] M. Chasles. Note sur les propriétés générales du système de deux corps semblables entr'eux et placés d'une manière quelconque dans l'espace; et sur le déplacement fini ou infiniment petit d'un corps solide libre. *Bulletin des Sciences Mathématiques*, pages 321–326, 1830.

[26] D. Zlatanov, I.A. Bonev, and C.M. Gosselin. Constraint singularities as $\mathcal{C}$-space singularities. *Advances in Robot Kinematics (ARK'02)*, pages 183–192, 2002.

[27] H.B. Choi, O. Campany, F. Pierrot, A. Konno, T. Shibukawa, and M. Uchiyama. Design and control of a novel 4-dofs parallel robot H4. *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, 2003.

[28] Erol Özgür. From lines to dynamics of parallel robots. *Thèse de Doctorat, Université Blaise Pascal*, 2012.

[29] R.C.P. Paul. Robot manipulators: mathematics, programming and control. *MIT Press*, 1981.

[30] J.Y.S. Luh, M.W. Walker, and R.C.P. Paul. On-line computational scheme for mechanical manipulators. *Transaction of ASME, J. of Dynamic Systems, Measurement, and Control*, pages 69–76, 1980.

[31] Jean le Rond d'Alembert. Traité de dynamique. 1743.