



HAL
open science

Recursive and Symbolic Calculation of the Elastodynamic Model of Flexible Parallel Robots

Sébastien Briot, Wisama Khalil

► **To cite this version:**

Sébastien Briot, Wisama Khalil. Recursive and Symbolic Calculation of the Elastodynamic Model of Flexible Parallel Robots. *The International Journal of Robotics Research*, 2014, 33 (3), pp.469-483. hal-00859539

HAL Id: hal-00859539

<https://hal.science/hal-00859539v1>

Submitted on 24 Jun 2019

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

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

Recursive and Symbolic Calculation of the Elastodynamic Model of Flexible Parallel Robots

Sébastien Briot* and Wisama Khalil†

September 7, 2013

Abstract

This paper presents a symbolic and recursive calculation of the elastodynamic model of flexible parallel robots. In order to reduce the computational time required for simulating the elastodynamic behavior of robots, it is necessary to minimize the number of operators in the symbolic expression of the model. Some algorithms have been proposed for the rigid case, for parallel robots with lumped springs or for serial robots with distributed flexibilities. In this paper, we extend the previous works to parallel robots with distributed flexibilities.

The generalized Newton-Euler model is used and combined with the principle of virtual powers to minimize the number of operators and intermediate variables. Recursive calculations are proposed for the computation of the Jacobian matrices defining the kinematic constraints in order to decrease the number of operators. The proposed algorithm is used to compute the elastodynamic model of a prototype of a planar parallel robot developed at IRCCyN: the DualEMPS. The computed model is compared both with simulations done on Adams and with experiments. The validity of the approach in terms of result accuracy and computational time is demonstrated.

1 Introduction

Many studies have been devoted to the computation of the full dynamic model of rigid parallel robots [Khalil and Ibrahim, 2007, Ibrahim and Khalil, 2010, Moon, 2007], however there are still many opened questions concerning the computation of their elastodynamic model. One of them concerns

*Institut de Recherches en Communications et Cybernétique de Nantes, UMR CNRS 6597, 1 rue de la Noë, 44321 Nantes Cedex, France, Tel: +33 (0)2 40 37 69 58, Fax: +33 (0)2 40 37 69 30, Sebastien.Briot@ircyn.ec-nantes.fr

†LUNAM University, Ecole Centrale de Nantes and Institut de Recherches en Communications et Cybernétique de Nantes, UMR CNRS 6597, 1 rue de la Noë, 44321 Nantes Cedex, France, Tel: +33 (0)2 40 37 69 46, Fax: +33 (0)2 40 37 69 30, Wisama.Khalil@ircyn.ec-nantes.fr

the reduction of the computational time that is generally huge and prevents the use of such models in many applications, such as real-time control, design optimization process, etc. To decrease the computational cost, it is either (i) possible to decrease the number of variables (using model reduction methods [Craig and Bampton, 1968, Briot et al., 2011, Craig, 1981] and truncated series of shape functions [Blevins, 2001]) or (ii) to efficiently compute the symbolic model in order to minimize the number of operators (similarly to what has been done for rigid robots [Khalil and Dombre, 2002], robots with lumped springs [Khalil and Gautier, 2000] or for serial robots with distributed flexibilities [Boyer and Khalil, 1998]). Both methods can obviously be combined. However, this paper only focuses on the efficient symbolic computation for the elastodynamic model of parallel robots.

For the computation of the elastodynamic models of robots, two main approaches are generally proposed (see [Dwivedy and Eberhard, 2006] for a large literature review): (i) lumped modeling [Khalil and Gautier, 2000, Kruszewski et al., 1975, Wittbrodt et al., 2006] and (ii) modeling using distributed flexibilities [Boyer and Khalil, 1998, Shabana, 2005, Bauchau, 2011, Stachera and Schumacher, 2008, Rognant et al., 2010, de Jalon and Bayo]. The lumped modeling is generally simpler to use by non-experts in finite element methods but, to obtain a correct model accuracy, higher number of elements is required, thus increasing the computational time. The most relevant works in lumped modeling methods are probably [Khalil and Gautier, 2000] and [Wittbrodt et al., 2006]. In [Khalil and Gautier, 2000], the flexibilities are modeled by one degree of freedom (*dof*) springs and a systematic procedure for the symbolic computation of the model is proposed. This procedure allows the minimization of the number of operators in the model. In [Wittbrodt et al., 2006], springs of higher dimension are used, but it is shown that for obtaining good accuracy, the number of elements must be high, thus leading to bigger computational time.

Contrary to lumped modeling, using distributed flexibilities allows the improvement of the model accuracy. However, such methods require highly-skilled users. In [Shabana, 2005, Bauchau, 2011, Rognant et al., 2010, de Jalon and Bayo], some general methodologies based on the Lagrange principle that can be applied to any system are proposed. In the case of closed-loop mechanisms, some Jacobian matrices are computed that allow taking into account the kinematic dependencies. The work [Stachera and Schumacher, 2008] combines the Lagrange principle and the principle of virtual works for computing the elastodynamic model of parallel robots. However, the main drawback of such general methodologies is that they are not specifically designed for parallel robots and that they do not guarantee the minimization of the number of operators for the symbolic computation of the model. A first approach for the systematic computation of the required Jacobian matrices has been proposed in [Bouzgarrou et al., 2005]. However, this approach was not designed so that a minimal number of operators for

the model symbolic computation can be obtained. Indeed, to the best of our knowledge, a systematic procedure to compute the elastodynamic model (using distributed flexibilities) of parallel robot with a minimal numbers of operators has never been proposed, even if such models are useful for several different reasons:

- In design optimization processes, optimization algorithms that test thousands of robot parameters are used. If the computational time required for the calculation of one iteration of the the elastodynamic model of the robot is not minimized, several days, and even month, can pass before the results are obtained.
- Symbolic expressions, with a minimized number of variables and operators, are requested for computing the identification model, in order to decrease the risk of error propagation due to the noisy measured data.

The present work aims at filling this gap. In order to minimize the number of operations, a generalized Newton-Euler (*NE*) model (which is known to reduce the number of operators [Khalil and Dombre, 2002, Boyer and Khalil, 1998, Boyer et al., 2007, Shabana, 1990]) is used and combined with the principle of virtual powers (*PVP*). The Jacobian matrices defined in the *PVP* are computed using recursive algorithms that decrease the number of operators. For computing the full elastodynamic model of parallel robots, the approach proposed in [Khalil and Ibrahim, 2007, Ibrahim and Khalil, 2010] for rigid robots and in [Stachera and Schumacher, 2008] for flexible robots is adapted. This approach proposes to:

1. convert the parallel robot into a virtual system defined by: (i) a tree-structure robot composed of the kinematic chains of the actual robot for which all joints (passive and active) are considered actuated and (ii) a free body (the platform which is considered as rigid) (Fig. 1),
2. compute the elastodynamic model of this new virtual system,
3. finally, close the loops by using the *PVP*.

This method is effective, systematic and can be applied to any parallel robot.

The paper is organized as follows. In Section 2, the computation of the generalized *NE* model of a flexible free body is recalled. Then, in Section 3, the elastodynamic model of the virtual tree structure is developed. Section 4 shows the computation of the elastodynamic model of the actual parallel robot. In Section 5, the proposed approach is applied for the computation of the elastodynamic model of a prototype of parallel robot designed at the IRCCyN, named the DualEMPS. Finally, in Section 6, conclusions are drawn.

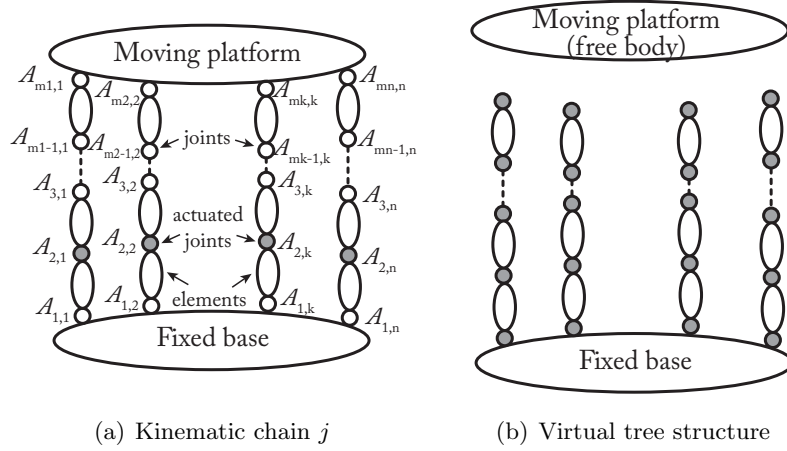


Figure 1: A general parallel robot.

A first condensed version of this work has been proposed in [Briot and Khalil, 2013]. The present paper contains detailed explanations to enlighten the theoretical understanding of the method, especially in Section 4 in which, contrary to [Briot and Khalil, 2013] that focused on isostatic robots, a general and systematic method for the choice of the generalized coordinates of overconstrained parallel robots is presented. Moreover, this is the first time that experimental validations are shown.

2 Recall on the Generalized Newton-Euler Model of a Flexible Free Body

This section aims at making some recalls on the way to obtain the generalized *NE* model of a flexible free body. For further development, the reader is referred to [Boyer and Khalil, 1998, Boyer et al., 2007, Shabana, 1990, Sharf and Damaren, 1992].

2.1 Kinematics of a Flexible Free Body

The twist of any free flexible body j at any point M_j (Fig. 2(a)) can always be expressed as:

$$\begin{bmatrix} \mathbf{v}_j(M_j) \\ \omega_j(M_j) \end{bmatrix} = \mathbf{t}_j + \begin{bmatrix} \omega_j(A_j) \times \mathbf{r}_j(M_j) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{e_j}(M_j) \\ \omega_{e_j}(M_j) \end{bmatrix} \quad (1)$$

where $\mathbf{t}_j = \left[\mathbf{v}_j^T(A_j), \omega_j^T(A_j) \right]^T$ is the twist of the local frame fixed on the body j expressed at point A_j , $\mathbf{v}_j(A_j)$ and $\omega_j(A_j)$ being the translational and rotational velocities, respectively, $\mathbf{r}_j(M_j)$ is the position vector of the

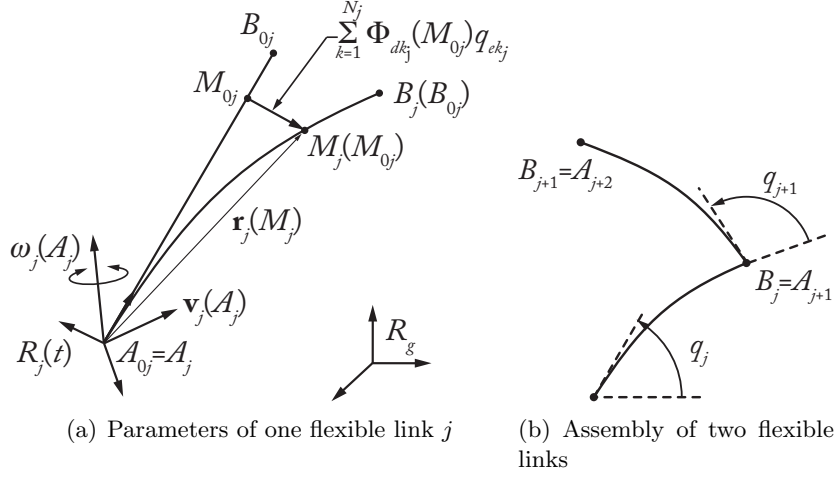


Figure 2: Schematics of the flexible elements into consideration.

point M_j (of the deformed body) with respect to the local frame, $\mathbf{v}_{e_j}(M_j)$ and $\omega_{e_j}(M_j)$ are the translational and rotational velocities due to the body elasticity that can be parameterized as truncated series of Rayleigh-Ritz shape functions:

$$\begin{bmatrix} \mathbf{v}_{e_j}(M_j) \\ \omega_{e_j}(M_j) \end{bmatrix} = \begin{bmatrix} \Phi_{d_j}(M_{0j}) \\ \Phi_{r_j}(M_{0j}) \end{bmatrix} \dot{\mathbf{q}}_{e_j} = \Phi_j(M_{0j}) \dot{\mathbf{q}}_{e_j} \quad (2)$$

with $\Phi_{d,r_j} = [\phi_{d,r1_j}, \dots, \phi_{d,rN_{j_j}}]$, $\phi_{dk_j}(M_{0j})$ and $\phi_{rk_j}(M_{0j})$ being the k -th shape functions for the displacement and the rotation of the flexible body expressed at point M_{0j} , respectively, and $\dot{\mathbf{q}}_{e_j}^T = [\dot{q}_{e1_j}, \dots, \dot{q}_{eN_{j_j}}]$, \dot{q}_{ek_j} being the k -th elastic generalized velocity of the body and N_j the number of considered shape functions. It should be noted that the vector $\mathbf{r}_j(M_j)$ of (1) can be expressed as:

$$\mathbf{r}_j(M_j) = \mathbf{r}_j(M_{0j}) + \Phi_{d_j}(M_{0j}) \mathbf{q}_{e_j} \quad (3)$$

where $\mathbf{r}_j(M_{0j})$ is the position of the point M_{0j} with respect to the local frame and $\mathbf{q}_{e_j}^T = [q_{e1_j}, \dots, q_{eN_{j_j}}]$ are the elastic generalized coordinates of the body.

Equations (1) to (3) define the kinematic model of the flexible free body j . This model is thus parameterized by the following set of variables:

- \mathbf{t}_j that are the Euler variables characterizing the rigid displacement of the body j at the origin of the local frame,
- \mathbf{q}_{e_j} that are the Lagrange variables characterizing the elastic displacement of the body j .

Thus, the kinematics model of a flexible free body is parameterized by N_j supplementary variables \mathbf{q}_{e_j} and N_j shape functions compared to the free rigid body case.

It should be mentioned that this description can be applied to both robot segments and joints, as long as all the shape functions can be defined.

2.2 Matrix Form of the Generalized Newton-Euler Model for a Flexible Free Body

Skipping all mathematical derivations and referring the reader to [Boyer and Khalil, 1998, Boyer et al., 2007, Shabana, 1990, Sharf and Damaren, 1992], the generalized *NE* model of a flexible free body can be obtained via the application of the *PVP*. This model takes the form:

$$\begin{aligned} \begin{bmatrix} \Delta \mathbf{f}_{c_j} \\ \Delta \mathbf{c}_{c_j} \\ \Delta \mathbf{s}_{c_j} \end{bmatrix} &= \begin{bmatrix} m_j \mathbf{I}_{d3} & \widehat{\mathbf{MS}}_j^T & \mathbf{MS}_{de_j} \\ \widehat{\mathbf{MS}}_j & \mathbf{I}_j & \mathbf{MS}_{re_j} \\ \mathbf{MS}_{de_j}^T & \mathbf{MS}_{re_j}^T & \mathbf{M}_{ee_j} \end{bmatrix} \begin{bmatrix} \gamma_j(A_j) \\ \alpha_j(A_j) \\ \ddot{\mathbf{q}}_{e_j} \end{bmatrix} + \\ \begin{bmatrix} \mathbf{f}_{in_j} \\ \mathbf{c}_{in_j} \\ \mathbf{s}_{in_j} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{K}_{ee_j} \mathbf{q}_{e_j} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{g_j} \\ \mathbf{c}_{g_j} \\ \mathbf{s}_{g_j} \end{bmatrix} &= \mathbf{M}_j \begin{bmatrix} \dot{\mathbf{t}}_j \\ \ddot{\mathbf{q}}_{e_j} \end{bmatrix} + \mathbf{c}_j \end{aligned} \quad (4)$$

where

- m_j the total mass of the body j
- \mathbf{I}_{dk} is the $k \times k$ Identity matrix,
- \mathbf{I}_j is the 3×3 total inertia matrix of the body j ,
- $\widehat{\mathbf{MS}}_j$ is the 3×3 matrix containing the first moments of inertia of the body j ,
- \mathbf{M}_{ee_j} is the $N_j \times N_j$ elastic mass matrix of the body j ,
- \mathbf{MS}_{de_j} and \mathbf{MS}_{re_j} are $3 \times N_j$ matrices,
- $\dot{\mathbf{t}}_j = [\gamma_j(A_j)^T, \alpha_j(A_j)^T]^T$ is the acceleration of frame j expressed at point A_j with $\gamma_j(A_j)$ and $\alpha_j(A_j)$ the translational and rotational accelerations of the local frame fixed on the body j at point A_j , respectively,
- \mathbf{f}_{in_j} and \mathbf{c}_{in_j} are vectors of the inertial force and torques, respectively,
- \mathbf{s}_{in_j} is the vector of the generalized elastic forces,
- \mathbf{f}_{g_j} and \mathbf{c}_{g_j} are vectors of the gravity force and torques plus the other external forces, respectively,
- \mathbf{s}_{g_j} is the vector of the generalized elastic forces due to gravity,

- \mathbf{K}_{ee_j} the stiffness matrix of the body j ,
- \mathbf{M}_j is the global mass matrix of the body j ,
- \mathbf{c}_j is the global vector of the centrifugal, Coriolis, gravity and elastic forces of the body j ,
- $\Delta\mathbf{f}_{c_j}$ and $\Delta\mathbf{c}_{c_j}$ are the total sum of forces and torques transmitted by the joints, respectively,
- $\Delta\mathbf{s}_{c_j}$ is the total sum of the elastic generalized forces transmitted by the joints.

The computation of all these terms is detailed in the Appendix.

This generalized *NE* model is known to reduce the number of operators necessary for the computation of the elastodynamics behavior of a flexible link [Boyer and Khalil, 1998].

3 Computation of the Elastodynamic Model for the Virtual System

Let us consider a parallel robot composed of one rigid fixed base (denoted as the element 0), one rigid moving platform and n legs, each leg being a serial kinematic chain composed of $m_i - 1$ elements linked by m_i joints (revolute, prismatic or even fixed joints - $i = 1, \dots, n$) (Fig. 1(a)). The actuated variables are denoted by \mathbf{q}_a and the leg passive variables by \mathbf{q}_p . The platform coordinates are denoted as \mathbf{x}_p . The size n_a of \mathbf{q}_a must be equal or superior to the number of *dof* of the parallel robot. The number of shape functions by element is denoted as N_{ij} ($j = 1, \dots, m_i - 1$). As a result, there are $n_e = \sum_{i=1}^n \sum_{j=1}^{m_i-1} N_{ij}$ elastic variables grouped in the vector \mathbf{q}_e . All the active, passive and elastic variables are grouped into the vector $\mathbf{q}_t^T = [\mathbf{q}_a^T, \mathbf{q}_p^T, \mathbf{q}_e^T]$.

3.1 Application of the PVP

Considering the link j of leg i (denoted in the following as the element ij), the *PVP* states that:

$$\begin{bmatrix} \mathbf{t}_{ij}^{*T} & \dot{\mathbf{q}}_{e_{ij}}^{*T} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{f}_{c_{ij}}^T & \Delta\mathbf{c}_{c_{ij}}^T & \Delta\mathbf{s}_{c_{ij}}^T \end{bmatrix}^T = \dot{\mathbf{q}}_t^{*T} \begin{bmatrix} \tau_{t_{ij}} \\ \mathbf{0}_{n_e} \end{bmatrix} \quad (5)$$

where $\tau_{t_{ij}}$ is the vector of the virtual input torques of the tree structure (Fig. 1(b)) due to the movement of the link ij and $\mathbf{0}_{n_e}$ a null vector of dimension n_e .

The twist \mathbf{t}_{ij}^* and generalized elastic velocities $\dot{\mathbf{q}}_{e_{ij}}^*$ are linked to the generalized velocities $\dot{\mathbf{q}}_t^*$ by the relation:

$$\begin{bmatrix} \mathbf{t}_{ij} \\ \dot{\mathbf{q}}_{e_{ij}} \end{bmatrix} = \mathbf{J}_{ij} \dot{\mathbf{q}}_t \quad (6)$$

where \mathbf{J}_{ij} is the Jacobian matrix of the element ij whose expression will be given in the following section.

Eq. (5) can thus be rewritten as:

$$\dot{\mathbf{q}}_t^{*T} \mathbf{J}_{ij}^T \begin{bmatrix} \Delta \mathbf{f}_{c_{ij}}^T & \Delta \mathbf{c}_{c_{ij}}^T & \Delta \mathbf{s}_{c_{ij}}^T \end{bmatrix}^T = \dot{\mathbf{q}}_t^{*T} \begin{bmatrix} \tau_{t_{ij}} \\ \mathbf{0}_{n_e} \end{bmatrix} \quad (7)$$

which leads to, for any virtual velocity $\dot{\mathbf{q}}_t^*$:

$$\begin{bmatrix} \tau_{t_{ij}} \\ \mathbf{0}_{n_e} \end{bmatrix} = \mathbf{J}_{ij}^T \begin{bmatrix} \Delta \mathbf{f}_{c_{ij}}^T & \Delta \mathbf{c}_{c_{ij}}^T & \Delta \mathbf{s}_{c_{ij}}^T \end{bmatrix}^T \quad (8)$$

Thus, now considering all the links of the robot, it comes that

$$\begin{aligned} \begin{bmatrix} \tau_t = \sum_{i,j} \tau_{t_{ij}} \\ \mathbf{0}_{n_e} \end{bmatrix} &= \sum_{i,j} \mathbf{J}_{ij}^T \begin{bmatrix} \Delta \mathbf{f}_{c_{ij}} \\ \Delta \mathbf{c}_{c_{ij}} \\ \Delta \mathbf{s}_{c_{ij}} \end{bmatrix} \\ &= \sum_{i,j} \mathbf{J}_{ij}^T \left(\mathbf{M}_{ij} \begin{bmatrix} \dot{\mathbf{t}}_{ij} \\ \ddot{\mathbf{q}}_{e_{ij}} \end{bmatrix} + \mathbf{c}_{ij} \right) \end{aligned} \quad (9)$$

where τ_t is the vector of the tree-structure input efforts.

In the next section, recursive algorithms for the computation of the vectors \mathbf{t}_{ij} , $\dot{\mathbf{t}}_{ij}$ and of the Jacobian matrices \mathbf{J}_{ij} are developed.

3.2 Recursive Computation of the Velocities and Jacobian Matrices

Let us consider the Fig. 2 describing the displacement of the element j . From (1), it comes that

$$\begin{bmatrix} \mathbf{v}_j(B_j) \\ \omega_j(B_j) \end{bmatrix} = \mathbf{t}_j + \begin{bmatrix} \omega_j(A_j) \times \mathbf{r}_j(B_j) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{e_j}(B_j) \\ \omega_{e_j}(B_j) \end{bmatrix} \quad (10)$$

If an element $j + 1$ is linked at B_j by an actuated joint moving at a velocity \dot{q}_{j+1} (Fig. 2(b) – if the joint is a fixed joint, $\dot{q}_{j+1} = 0$), it comes that:

$$\mathbf{t}_{j+1} = \begin{bmatrix} \mathbf{v}_j(B_j) \\ \omega_j(B_j) \end{bmatrix} + \dot{q}_{j+1} \mathbf{a}_{j+1} \quad (11)$$

where \mathbf{a}_{j+1} is the unit twist describing the motion of the actuator [Khalil and Dombre, 2002].

As a result, for the element ij of the global robot, it can be demonstrated that (in the following expressions, the preceding superscript indicate the frame in which the vector expression is given) [Boyer and Khalil, 1998]:

$$\begin{aligned} {}^{ij}\mathbf{t}_{ij} &= {}^{ij}\mathbf{T}_{i(j-1)} {}^{i(j-1)}\mathbf{t}_{i(j-1)} + \\ & {}^{ij}\overline{\mathbf{R}}_{i(j-1)} \Phi_{i(j-1)}(A_{ij}) \dot{\mathbf{q}}_{e_{i(j-1)}} + \dot{q}_{ij} {}^{ij}\mathbf{a}_{ij} \end{aligned} \quad (12)$$

which can also be written as:

$${}^{ij}\mathbf{t}_{ij} = \mathbf{J}_{t_{ij}} \dot{\mathbf{q}}_t \quad (13)$$

with

$$\mathbf{J}_{t_{ij}} = {}^{ij}\mathbf{T}_{i(j-1)} \mathbf{J}_{t_{i(j-1)}} + \Phi_{\mathbf{q}_{e_{ij}}} + \mathbf{A}_{ij} \quad (14)$$

where

$$\begin{aligned} \Phi_{\mathbf{q}_{e_{ij}}} &= [\mathbf{0} \quad \cdots \quad {}^{ij}\overline{\mathbf{R}}_{i(j-1)} \Phi_{i(j-1)}(A_{ij}) \quad \cdots \quad \mathbf{0}] \\ \mathbf{A}_{ij} &= [\mathbf{0} \quad \cdots \quad {}^{ij}\mathbf{a}_{ij} \quad \cdots \quad \mathbf{0}] \end{aligned} \quad (15)$$

In the matrix $\Phi_{\mathbf{q}_{e_{ij}}}$, the term ${}^{ij}\overline{\mathbf{R}}_{i(j-1)} \Phi_{i(j-1)}(A_{ij})$ is located at the columns corresponding to the variables $\dot{\mathbf{q}}_{e_{i(j-1)}}$ and, for the matrix \mathbf{A}_{ij} , the term ${}^{ij}\mathbf{a}_{ij}$ is located at the column corresponding to the variable \dot{q}_{ij} .

In the previous expressions, matrix $\Phi_{i(j-1)}$ is the matrix containing all shape functions for the element $i(j-1)$ and:

$$\begin{aligned} {}^{ij}\overline{\mathbf{R}}_{i(j-1)} &= \begin{bmatrix} {}^{ij}\mathbf{R}_{i(j-1)} & \mathbf{0} \\ \mathbf{0} & {}^{ij}\mathbf{R}_{i(j-1)} \end{bmatrix} \\ {}^{ij}\mathbf{T}_{i(j-1)} &= {}^{ij}\overline{\mathbf{R}}_{i(j-1)} \left(\mathbf{I}_{d6} - \begin{bmatrix} \mathbf{0} & {}^{i(j-1)}\hat{\mathbf{r}}_{i(j-1)}(A_{ij}) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) \end{aligned} \quad (16)$$

where ${}^{ij}\mathbf{R}_{i(j-1)}$ is the rotation matrix between frames ij and $i(j-1)$, ${}^{i(j-1)}\hat{\mathbf{r}}_{i(j-1)}(A_{ij})$ is the cross product matrix associated with the vector ${}^{i(j-1)}\mathbf{r}_{i(j-1)}(A_{ij})$, i.e the position of point A_{ij} in the frame $i(j-1)$.

Finally, the global Jacobian matrix \mathbf{J}_{ij} of (6) can be computed as:

$$\mathbf{J}_{ij} = \begin{bmatrix} \mathbf{J}_{t_{ij}} \\ \mathbf{O}_{\mathbf{q}_{e_{ij}}} \end{bmatrix} \quad (17)$$

where $\mathbf{O}_{\mathbf{q}_{e_{ij}}}$ is defined such that

$$\dot{\mathbf{q}}_{e_{ij}} = \mathbf{O}_{\mathbf{q}_{e_{ij}}} \dot{\mathbf{q}}_e$$

3.3 Recursive Computation of the Accelerations

Differentiating (10), it can be shown that [Boyer and Khalil, 1998]:

$$\begin{aligned} {}^{ij}\dot{\mathbf{t}}_{ij} = & {}^{ij}\mathbf{T}_{i(j-1)} {}^{i(j-1)}\dot{\mathbf{t}}_{i(j-1)} + \ddot{q}_{ij} {}^{ij}\mathbf{a}_{ij} + {}^{ij}\mathbf{h}_{ij} + \\ & {}^{ij}\overline{\mathbf{R}}_{i(j-1)} \Phi_{i(j-1)}(A_{ij}) \ddot{\mathbf{q}}_{e_{i(j-1)}} \end{aligned} \quad (18)$$

where:

$$\begin{aligned} {}^{ij}\mathbf{h}_{ij} &= \begin{bmatrix} {}^{i(j-1)}\mathbf{h}_{lin_{ij}} \\ {}^{i(j-1)}\mathbf{h}_{ang_{ij}} \end{bmatrix} \\ {}^{ij}\mathbf{h}_{lin_{ij}} &= {}^{ij}\mathbf{R}_{i(j-1)} {}^{i(j-1)}\omega_{i(j-1)}(A_{i(j-1)}) \times \\ & \quad {}^{ij}\mathbf{R}_{i(j-1)} (2\Phi_{d_{i(j-1)}}(A_{ij}) \dot{\mathbf{q}}_{e_{i(j-1)}} + \\ & \quad {}^{i(j-1)}\omega_{i(j-1)}(A_{i(j-1)}) \times {}^{i(j-1)}\mathbf{r}_{i(j-1)}(A_{ij})) \\ {}^{ij}\mathbf{h}_{ang_{ij}} &= {}^{ij}\mathbf{R}_{i(j-1)} (\dot{q}_{ij} ({}^{i(j-1)}\omega_{i(j-1)}(A_{i(j-1)}) + \\ & \quad \Phi_{r_{i(j-1)}}(A_{ij}) \dot{\mathbf{q}}_{e_{i(j-1)}}) \times {}^{ij}\mathbf{a}_{ij} + \\ & \quad {}^{i(j-1)}\omega_{i(j-1)}(A_{i(j-1)}) \times \\ & \quad {}^{ij}\mathbf{R}_{i(j-1)} \Phi_{r_{i(j-1)}}(A_{ij}) \dot{\mathbf{q}}_{e_{i(j-1)}}) \end{aligned} \quad (19)$$

Eq. (18) can be then put into the form:

$${}^{ij}\dot{\mathbf{t}}_{ij} = \mathbf{J}_{t_{ij}} \ddot{\mathbf{q}}_t + {}^{ij}\mathbf{g}_{ij} \quad (20)$$

with

$${}^{ij}\mathbf{g}_{ij} = {}^{ij}\mathbf{h}_{ij} + {}^{ij}\mathbf{T}_{i(j-1)} {}^{i(j-1)}\mathbf{g}_{i(j-1)} \quad (21)$$

initialized with ${}^{i0}\mathbf{g}_{i0} = \mathbf{0}$ if the base is fixed. Thus,

$$\begin{bmatrix} {}^{ij}\dot{\mathbf{t}}_{ij} \\ \ddot{\mathbf{q}}_{e_{ij}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{t_{ij}} \\ \mathbf{O}_{q_{e_{ij}}} \end{bmatrix} \ddot{\mathbf{q}}_t + \begin{bmatrix} {}^{ij}\mathbf{g}_{ij} \\ \mathbf{0} \end{bmatrix} = \mathbf{J}_{ij} \ddot{\mathbf{q}}_t + {}^{ij}\mathbf{g}_{ij}^s \quad (22)$$

3.4 Elastodynamic model of the virtual system

Introducing (22) into (9) leads to:

$$\begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} = \sum_{i,j} \mathbf{J}_{ij}^T \mathbf{M}_{ij} \mathbf{J}_{ij} \ddot{\mathbf{q}}_t + \mathbf{c}_{ij}^s \quad (23)$$

where

$$\mathbf{c}_{ij}^s = \mathbf{J}_{ij}^T (\mathbf{c}_{ij} + \mathbf{M}_{ij} {}^{ij}\mathbf{g}_{ij}^s) \quad (24)$$

The *NE* equations for the rigid moving platform are given by [Ibrahim and Khalil, 2010]:

$$\mathbf{f}_p = \mathbf{M}_p \dot{\mathbf{t}}_p + \mathbf{c}_p \quad (25)$$

where \mathbf{f}_p is the platform reaction wrench expressed at the platform local frame origin, \mathbf{M}_p is the platform mass matrix, $\dot{\mathbf{t}}_p$ is the platform acceleration screw and \mathbf{c}_p the centrifugal, Coriolis, gravity effects and external efforts applied on the platform.

Finally, the global elastodynamic model of the virtual structure can be put into the form:

$$\begin{aligned} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \\ \mathbf{f}_p \end{bmatrix} &= [\sum_{i,j} \mathbf{J}_{ij}^T \mathbf{M}_{ij} \mathbf{J}_{ij} \quad \mathbf{M}_p] \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \begin{bmatrix} \mathbf{c}_{ij}^s \\ \mathbf{c}_p \end{bmatrix} \\ &= \mathbf{M}_t \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{c}_t \end{aligned} \quad (26)$$

Adding the contributions of the motor inertia and friction effects [Khalil and Dombre, 2002]:

$$\begin{aligned} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \\ \mathbf{f}_p \end{bmatrix} &= \mathbf{M}_t^s \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{c}_t^s + \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \\ \begin{bmatrix} \mathbf{F}_v \dot{\mathbf{q}}_t \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_s \text{sign}(\dot{\mathbf{q}}_t) \\ \mathbf{0} \end{bmatrix} &= \mathbf{M}_{tot} \ddot{\mathbf{q}}_{tot} + \mathbf{c}_{tot} \end{aligned} \quad (27)$$

where \mathbf{I}_t is a diagonal matrix whose j -th element corresponds to the value of the inertia of joint j (the j -th element of \mathbf{I}_t is equal to zero if the joint is passive or if it corresponds to an elastic coordinate) and \mathbf{F}_v , (\mathbf{F}_s , resp.) a diagonal matrix of viscous (Coulomb, resp.) friction parameters.

4 Computation of the Elastodynamic Model of Parallel Robots

The model of the virtual tree structure and of the free moving platform does not consider the closed-loop kinematic chains. As a matter of fact, the $n_{q_{tot}}$ components of the generalized velocity vector $\dot{\mathbf{q}}_{tot}^T = [\dot{\mathbf{q}}_t^T \quad \dot{\mathbf{t}}_p^T]$ are dependent. The independent components are gathered in vector $\dot{\mathbf{q}}$ ($\dim \dot{\mathbf{q}} = n_q < n_{tot}$) and their determination is described thereafter.

4.1 Determination of the Generalized Coordinates of the Parallel Robot

For determining one possible subset of generalized coordinates for the parallel robot, let us express the relations between the vector of generalized velocities of the tree structure $\dot{\mathbf{q}}_t$ and the twist of the last element m_i for each leg i . Using (6) for computing the twist ${}^{i,m_i}\mathbf{t}_{i,m_i}$ of the extremity of each leg, it comes that:

$${}^{i,m_i}\mathbf{t}_{i,m_i} = \mathbf{J}_{t_i,m_i}^i \dot{\mathbf{q}}_{t_i} \quad (28)$$

where \mathbf{J}_{t_i, m_i}^i can be obtained from \mathbf{J}_{t_i, m_i} by extracting the columns corresponding to the vector $\dot{\mathbf{q}}_{t_i}^T = [\mathbf{q}_{a_i}^T, \mathbf{q}_{p_i}^T, \mathbf{q}_{e_i}^T]$, i.e. the vector concatenating all active, passive and elastic variables of the leg i .

As the leg extremity is also linked to the rigid platform, its twist can be related to the platform twist \mathbf{t}_p via the rigid body displacement relation:

$${}^{i, m_i} \mathbf{t}_{t_i, m_i} = \mathbf{J}_p^i \mathbf{t}_p, \text{ where } \mathbf{J}_p^i = {}^{i, m_i} \bar{\mathbf{R}}_0 \begin{bmatrix} \mathbf{I}_3 & -{}^0 \hat{\mathbf{p}}_i \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \quad (29)$$

in which \mathbf{J}_p^i is a 6×6 matrix, ${}^0 \hat{\mathbf{p}}_i$ is the cross product matrix of vector ${}^0 \mathbf{p}_i$ that characterizes the position of the attachment point A_{i, m_i} with respect to the platform center position (Fig. 1(a)) and ${}^{i, m_i} \bar{\mathbf{R}}_0$ is the 6×6 rotation matrix between the global frame and the local frame attached to element i, m_i .

Thus, expressing the twist ${}^{i, m_i} \mathbf{t}_{t_i, m_i}$ for each leg in relation with the platform twist \mathbf{t}_p and generalized coordinates $\dot{\mathbf{q}}_{t_i}$, the following set of equations is obtained:

$$\begin{bmatrix} \mathbf{J}_{t_1, m_1}^1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{t_n, m_n}^n \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{t_1} \\ \vdots \\ \dot{\mathbf{q}}_{t_n} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_p^1 \\ \vdots \\ \mathbf{J}_p^n \end{bmatrix} \mathbf{t}_p = \mathbf{0} \quad (30)$$

$$\Leftrightarrow \mathbf{J}_t \dot{\mathbf{q}}_t - \mathbf{J}_p \mathbf{t}_p = [\mathbf{J}_t \quad -\mathbf{J}_p] \begin{bmatrix} \dot{\mathbf{q}}_t \\ \mathbf{t}_p \end{bmatrix} = \mathbf{J}_{tot} \dot{\mathbf{q}}_{tot} = \mathbf{0}$$

where \mathbf{J}_{tot} is a $r \ n \times n_{q_{tot}}$ matrix, $n_{q_{tot}} > r \ n$ ($r = 6$ for a spatial robot, $r = 3$ for a planar robot). This means that a subset $\dot{\mathbf{q}}_d$ of $r \ n$ variables in vector $\dot{\mathbf{q}}_{tot}$ is linked to the others. This subset is not unique. An idea could be to put all passive joints and platform variables in this subset, i.e., $\dot{\mathbf{q}}_d^* = [\dot{\mathbf{q}}_p^T \ \mathbf{t}_p^T]^T$. However, for over-constrained parallel robots, $\dim(\dot{\mathbf{q}}_d^*) < r \ n$. As a result, this vector should be completed using some other elastic variables that could be chosen arbitrarily. Meanwhile, it must be mentioned that most of parallel robots have identical legs and that such a methodology will lead to an asymmetrical description of the leg variables, which is not ideal. In order to avoid this problem, we had better put in $\dot{\mathbf{q}}_d$ the last r components $\dot{\mathbf{q}}_{t_i}^f$ of each vector $\dot{\mathbf{q}}_{t_i}$ that is now decomposed into two parts: $\dot{\mathbf{q}}_{t_i}^T = [\dot{\mathbf{q}}_{t_i}^{0T} \ \dot{\mathbf{q}}_{t_i}^{fT}]$. Thus, variables $\dot{\mathbf{q}}_{t_i}^f$ are related to the others using (30):

$$\begin{aligned} & - \begin{bmatrix} \mathbf{J}_{t_1, m_1}^{f1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{t_n, m_n}^{fn} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^f \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^f \end{bmatrix} \\ & = \begin{bmatrix} \mathbf{J}_{t_1, m_1}^{01} & \cdots & \mathbf{0} & -\mathbf{J}_p^1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{t_n, m_n}^{0n} & -\mathbf{J}_p^n \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^0 \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^0 \\ \mathbf{t}_p \end{bmatrix} \end{aligned} \quad (31)$$

which can be written under the form

$$-\mathbf{J}_t^f \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^f \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^f \end{bmatrix} = [\mathbf{J}_t^0 \quad -\mathbf{J}_p] \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^0 \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^0 \\ \mathbf{t}_p \end{bmatrix} \quad (32)$$

or also

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^f \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^f \end{bmatrix} &= \dot{\mathbf{q}}_d = - \left(\mathbf{J}_t^f \right)^{-1} [\mathbf{J}_t^0 \quad -\mathbf{J}_p] \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^0 \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^0 \\ \mathbf{t}_p \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{J}_{d_{1,1}} & \cdots & \mathbf{J}_{d_{1,n}} & \mathbf{J}_{d_{1,n+1}} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{J}_{d_{n,1}} & \cdots & \mathbf{J}_{d_{n,n}} & \mathbf{J}_{d_{n,n+1}} \end{bmatrix} \dot{\mathbf{q}} \\ &= \mathbf{J}_d \dot{\mathbf{q}} \end{aligned} \quad (33)$$

where

- $\mathbf{J}_{t_i, m_i}^{0i}$ ($\mathbf{J}_{t_i, m_i}^{fi}$, resp.) are the columns of matrix \mathbf{J}_{t_i, m_i}^i corresponding to variables $\dot{\mathbf{q}}_{t_i}^0$ ($\dot{\mathbf{q}}_{t_i}^f$, resp.);
- $\mathbf{J}_{d_{ij}}$ is the matrix that relates the variable $\dot{\mathbf{q}}_{t_i}^f$ to $\dot{\mathbf{q}}_{t_j}^0$, $j = 1, \dots, n$;
- $\mathbf{J}_{d_{i, n+1}}$ is the matrix that relates the variable $\dot{\mathbf{q}}_{t_i}^f$ to \mathbf{t}_p .

It is noteworthy that the inversion of matrix \mathbf{J}_t^f involves only the inversion of the n $r \times r$ matrices $\mathbf{J}_{t_1, m_1}^{fj}$, which is more efficient in terms of computational time. Moreover, when 3D beam elements are used for leg i , if the coordinates $\mathbf{q}_{t_i}^f$ are the elastic coordinates of l -th element of this leg (denoted as element il), it can be proven that, as the k -th column of matrix $\mathbf{J}_{t_{il}}^{fi}$ corresponds to a unit twist that describes the displacement of the leg extremity due to the k -th coordinate of vector $\mathbf{q}_{t_i}^f$, $\mathbf{J}_{t_{il}}^{fi}$ is equal to [Khalil and Dombre, 2002]

$$\mathbf{J}_{t_{il}}^{fi} = \begin{bmatrix} {}^{i, m_i} \mathbf{R}_{il} & -{}^{i, m_i} \mathbf{R}_{il} {}^{il} \hat{\mathbf{p}}_{il} \\ \mathbf{0} & {}^{i, m_i} \mathbf{R}_{il} \end{bmatrix} \quad (34)$$

where ${}^{i, m_i} \mathbf{R}_{il}$ is the rotation matrix between the local frame linked at element i, m_i and the local frame linked at element il and ${}^{il} \hat{\mathbf{p}}_{il}$ is the cross product matrix of the vector ${}^{il} \mathbf{p}_{il}$ that characterizes the position of the leg extremity with respect to the frame linked at element il . Thus its inverse is equal to

$$\left(\mathbf{J}_{t_{il}}^{fi} \right)^{-1} = \begin{bmatrix} {}^{i, m_i} \mathbf{R}_{il}^T & \hat{\mathbf{p}}_{il} {}^{i, m_i} \mathbf{R}_{il}^T \\ \mathbf{0} & {}^{i, m_i} \mathbf{R}_{il}^T \end{bmatrix} \quad (35)$$

which does not require much calculation. If 2D beam elements are used, some similar relations can be obtained.

Finally, the Jacobian matrix relating all variables $\dot{\mathbf{q}}_{tot}$ to the configuration variables $\dot{\mathbf{q}}^T = [\dot{\mathbf{q}}_{t_1}^{0T} \cdots \dot{\mathbf{q}}_{t_n}^{0T} \mathbf{t}_p]$ can be obtained as:

$$\dot{\mathbf{q}}_{tot} = \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^0 \\ \dot{\mathbf{q}}_{t_1}^f \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^0 \\ \dot{\mathbf{q}}_{t_n}^f \\ \mathbf{t}_p \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{dc_1} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{J}_{d_{1,1}} & \cdots & \mathbf{J}_{d_{1,n}} & \mathbf{J}_{d_{1,n+1}} \\ \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{I}_{dc_n} & \mathbf{0} \\ \mathbf{J}_{d_{n,1}} & \cdots & \mathbf{J}_{d_{n,n}} & \mathbf{J}_{d_{n,n+1}} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I}_{d6} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_{t_1}^0 \\ \vdots \\ \dot{\mathbf{q}}_{t_n}^0 \\ \mathbf{t}_p \end{bmatrix} = \mathbf{J}\dot{\mathbf{q}} \quad (36)$$

where \mathbf{I}_{dc_i} is the $c_i \times c_i$ identity matrix, c_i being the size of vector $\dot{\mathbf{q}}_{t_i}^0$.

4.2 Computation of the Generalized Accelerations of the Actual Parallel Robot

Expressing the acceleration ${}^{i,m_i}\ddot{\mathbf{t}}_{i,m_i}$ of the extremity of each leg using (22) and combining this expression with the time derivative of (29), it comes that:

$${}^{i,m_i}\ddot{\mathbf{t}}_{i,m_i} = \mathbf{J}_{t_i,m_i}^i \ddot{\mathbf{q}}_{t_i} + {}^{i,m_i}\mathbf{g}_{i,m_i} = \mathbf{J}_p^i \dot{\mathbf{t}}_p + \dot{\mathbf{J}}_p^i \mathbf{t}_p \quad (37)$$

Thus, considering all the robot legs,

$$\begin{aligned} & \begin{bmatrix} \mathbf{J}_{t_1,m_1}^1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}_{t_n,m_n}^n \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_{t_1} \\ \vdots \\ \ddot{\mathbf{q}}_{t_n} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_p^1 \\ \vdots \\ \mathbf{J}_p^n \end{bmatrix} \dot{\mathbf{t}}_p \\ & + \begin{bmatrix} {}^{1,m_1}\mathbf{g}_{1,m_1} - \dot{\mathbf{J}}_p^1 \mathbf{t}_p \\ \vdots \\ {}^{n,m_n}\mathbf{g}_{n,m_n} - \dot{\mathbf{J}}_p^n \mathbf{t}_p \end{bmatrix} = \mathbf{0} \\ & \Leftrightarrow \mathbf{J}_t \ddot{\mathbf{q}}_t - \mathbf{J}_p \dot{\mathbf{t}}_p + \mathbf{b}^* = [\mathbf{J}_t \quad -\mathbf{J}_p] \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{b}^* = \mathbf{0} \end{aligned} \quad (38)$$

As a result, by analogy with (33), it can be demonstrated that

$$\ddot{\mathbf{q}}_d = \mathbf{J}_d \ddot{\mathbf{q}} + \left(\mathbf{J}_{t_{il}}^f \right)^{-1} \mathbf{b}^* = \mathbf{J}_d \ddot{\mathbf{q}} + \mathbf{b}_d \quad (39)$$

Finally, similarly to expression (36),

$$\ddot{\mathbf{q}}_{tot} = \begin{bmatrix} \ddot{\mathbf{q}}_{t_1}^0 \\ \ddot{\mathbf{q}}_{t_1}^f \\ \vdots \\ \ddot{\mathbf{q}}_{t_n}^0 \\ \ddot{\mathbf{q}}_{t_n}^f \\ \mathbf{t}_p \end{bmatrix} = \mathbf{J}\ddot{\mathbf{q}} + \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_{d1} \\ \vdots \\ \mathbf{0} \\ \mathbf{b}_{dn} \\ \mathbf{0} \end{bmatrix} = \mathbf{J}\ddot{\mathbf{q}} + \mathbf{b} \quad (40)$$

where \mathbf{b}_{di} is the part of the vector \mathbf{b}_d corresponding to the accelerations $\ddot{\mathbf{q}}_{ti}^f$.

4.3 Elastodynamic Model of the Actual Parallel Robot

Considering the actual robot, the *PVP* states that:

$$\dot{\mathbf{q}}_{tot}^{T*} (\mathbf{M}_{tot} \ddot{\mathbf{q}}_{tot} + \mathbf{c}_{tot}) = \dot{\mathbf{q}}_r^{T*} \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0}_{n_{eind}} \end{bmatrix} \quad (41)$$

where $\dot{\mathbf{q}}_r = \mathbf{E} \dot{\mathbf{q}}$, in which \mathbf{E} is a matrix that makes it possible to sort vector $\dot{\mathbf{q}}$ in such a manner that the first n_a rows of $\dot{\mathbf{q}}_r$ correspond to the vector $\dot{\mathbf{q}}_a$ of the actual active variables, $\boldsymbol{\tau}$ is the vector of the actual actuator input efforts and $\mathbf{0}_{n_{eind}}$ is a zero vector of dimension $n_{eind} = n_q - n_a$.

Introducing (36) into (41), it comes that,

$$\dot{\mathbf{q}}_r^{T*} \mathbf{E} \mathbf{J}^T (\mathbf{M}_{tot} \ddot{\mathbf{q}}_{tot} + \mathbf{c}_{tot}) = \dot{\mathbf{q}}_r^{T*} \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0}_{n_{eind}} \end{bmatrix} \quad (42)$$

(42) must be true for any value of $\dot{\mathbf{q}}_r^{T*}$, thus

$$\begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0}_{n_{eind}} \end{bmatrix} = \mathbf{E} \mathbf{J}^T (\mathbf{M}_{tot} \ddot{\mathbf{q}}_{tot} + \mathbf{c}_{tot}) \quad (43)$$

Finally, introducing (40) into (43) leads to:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0}_{n_{eind}} \end{bmatrix} &= \mathbf{E} \mathbf{J}^T \mathbf{M}_{tot} \mathbf{J} \mathbf{E}^T \ddot{\mathbf{q}}_r + \mathbf{E} \mathbf{J}^T (\mathbf{M}_{tot} \mathbf{b} + \mathbf{c}_{tot}) \\ &= \mathbf{M} \ddot{\mathbf{q}}_r + \mathbf{c} \end{aligned} \quad (44)$$

which is the full elastodynamic model of the parallel robot, with \mathbf{M} its mass matrix and \mathbf{c} the global vector of the centrifugal, Coriolis, gravity and elastic forces.

4.4 Discussion

In order to finally obtain the symbolic equations for the model with the minimum number of operations, the following method is used. First, the rigid kinematics of each element are modeled using the modified Denavit-Hartenberg notations [Khalil and Dombre, 2002]. If the link ij taken into consideration is flexible, N_{ij} supplementary elastic variables $\mathbf{q}_{e_{ij}}$ are introduced in combination with N_{ij} shape functions. Then, the previously developed equations are used in the following sequence:

- **Step 0: Initialization of the algorithm**
 Variables considered known: \mathbf{q} , $\dot{\mathbf{q}}$
 Computation of:

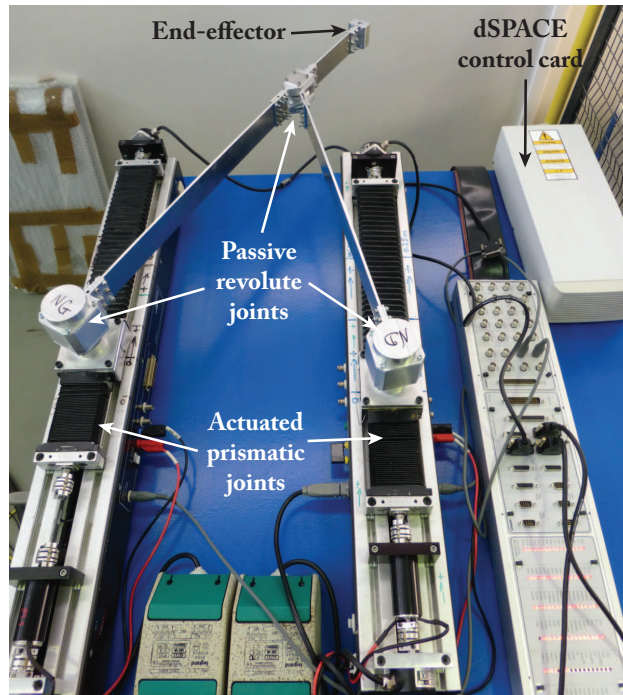
- $\Phi_{d_{i(j-1)}}(A_{ij}), \Phi_{r_{i(j-1)}}(A_{ij}), \Phi_{i(j-1)}(A_{ij})$ from (2); ${}^{i(j-1)}\mathbf{r}_{i(j-1)}(A_{ij})$ from (3);
 - $\mathbf{M}_{ij}, \mathbf{K}_{ee_{ij}}, \mathbf{f}_{g_{ij}}, \mathbf{c}_{g_{ij}}, \mathbf{s}_{g_{ij}}$ from (4);
 - $\mathbf{M}_p, \mathbf{c}_p$ from (25);
 - ${}^{ij}\overline{\mathbf{R}}_{i(j-1)}, {}^{ij}\overline{\mathbf{T}}_{i(j-1)}$ from (16); ${}^{ij}\mathbf{a}_{ij}$ from (11);
 - $\mathbf{A}_{ij}, \Phi_{\mathbf{q}_{eij}}$ from (15);
- **Step 1:** *Forward recurrence (computation of the twist and acceleration of each body, and computation of the Jacobian matrices required for applying the PVP on the virtual tree structure)*
Computation of:
- ${}^{ij}\mathbf{t}_{ij}$ from (12);
 - $\mathbf{f}_{in_{ij}}, \mathbf{c}_{in_{ij}}, \mathbf{s}_{in_{ij}}, \mathbf{c}_{ij}$ from (4);
 - $\mathbf{J}_{t_{ij}}$ from (14); \mathbf{J}_{ij} from (17);
 - ${}^{ij}\mathbf{h}_{ij}$ from (19); ${}^{ij}\mathbf{g}_{ij}$ from (21); \mathbf{g}_{ij}^s from (22);
- **Step 2:** *Computation of the global mass matrix and global vector of the centrifugal, Coriolis, gravity and elastic forces of the virtual tree structure*
Computation of:
- $\mathbf{M}_t, \mathbf{c}_t$ from (26);
 - $\mathbf{M}_{tot}, \mathbf{c}_{tot}$ from (27);
- **Step 3:** *Computation of the global mass matrix and global vector of the centrifugal, Coriolis, gravity and elastic forces of the parallel robot*
Computation of:
- $\mathbf{J}_t, \mathbf{J}_p$ from (30);
 - \mathbf{J}_t^f from (31); $(\mathbf{J}_t^f)^{-1}$ from (35);
 - \mathbf{J}_d from (33); \mathbf{J} from (36);
 - $\dot{\mathbf{J}}_p^i$ from (37);
 - \mathbf{b}^* from (38); \mathbf{b}_d from (39); \mathbf{b} from (40);
 - \mathbf{M}, \mathbf{c} from (44);
- **Step 4:** *Solving the model*
- Inverse dynamic model: computation of τ and $\ddot{\mathbf{q}}_e$ as a function of $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}_a$.
 - Forward dynamic model: computation of $\ddot{\mathbf{q}}$ as a function of $\mathbf{q}, \dot{\mathbf{q}}$ and τ .

For each computation, the elements of a vector or a matrix containing at least one mathematical operation are replaced by an intermediate variable. This variable is written in an output file which contains the model. The elements that do not contain any operations are not modified. The obtained vectors and matrices are propagated in the subsequent equations. Consequently, at the end, the dynamic model is obtained as a set of intermediate variables. Those that have no effect on the desired output (τ and the n_{eind} last values of $\ddot{\mathbf{q}}_r$ corresponding to the elastic variables in the case of the inverse model, $\ddot{\mathbf{q}}_r$ for the direct model) can be eliminated by scanning the intermediate variables from the end to the beginning. With this procedure, it is also possible to know the exact number of operators necessary for the computation of the model. This algorithm has been successfully implemented with Mathematica and is used in the next section for computing the elastodynamic model of a flexible planar parallel robot.

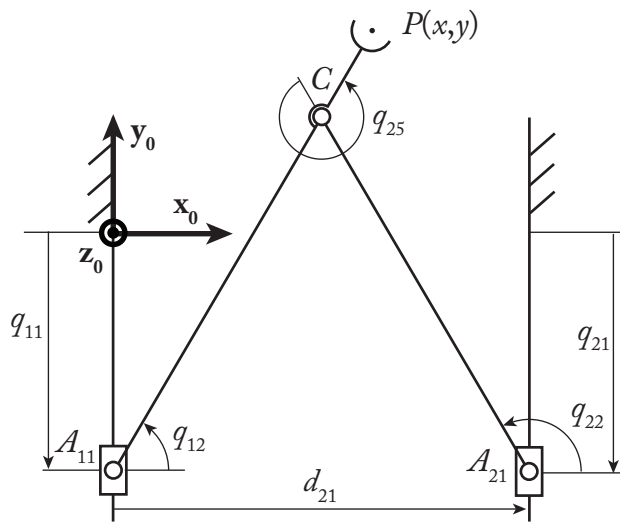
5 Case Study

The previous equations are used to compute the elastodynamic model of the DualEMPS, a prototype of flexible planar $\underline{P}RRR\underline{P}$ robot (Fig. 3 – R stands for a passive revolute joint and \underline{P} for an active prismatic joint) designed and manufactured at IRCCyN. This robot is actuated by two rotary actuators controlled by a dSPACE card in which simple PD control laws are introduced (the cut-off pulsation is set to 100 rad/s). The actuator movements are transmitted to the prismatic pairs via the use of ball screws (the stroke of the prismatic pairs is of 25cm). The two prismatic pairs are parallel and are linked to the two legs via passive revolute joints. Each leg is designed such that it can be decomposed as a succession of rigid and elastic links (Fig. 4) rigidly attached all together. The elastic links are very thin when compared to the rigid links. Their cross-sections are rectangular with a width of 4mm and an height of 50mm and they are made of Duralumin (AU4G). The two legs are connected via a passive revolute joint.

The modified Denavit-Hartenberg parameters corresponding to the architectural description of the Fig. 4 are given in Table 1, where the parameters d_{ij} and q_{ij} are detailed in Fig. 3. The gravity is directed along \mathbf{z}_0 . The *a priori* rigid dynamic parameters of the links have been extracted from CAD and are described in Table 2. In this table, m_{ij} is the mass of the link ij , mx_{ij} its first moment of inertia around the axes of the plane ($\mathbf{y}_{ij}O\mathbf{z}_{ij}$), zz_{ij} the second moment of inertia around \mathbf{z}_{ij} , fv_{ij} the viscous friction parameter and fs_{ij} the Coulomb inertia parameter. Note that the values of the friction terms have been set by hand so that the obtained results best fit with experimental data. The elastic links are modeled as planar beam finite elements (one element by elastic link, i.e. there are 9 elastic coordinates in total). Their corresponding elastic dynamic parameters are not detailed for



(a) Picture of the prototype



(b) Schematics of the kinematic chain

Figure 3: The DualEMPS flexible parallel robot.

reasons of paper compactness but can be easily computed using the formulas given in Section 2 applied to the beam shape functions given in [Shabana, 2005].

The model is thus calculated with Mathematica applying the proposed

Table 1: Modified Denavit-Hartenberg parameters of the DualEMPS robot.

ij	$a(ij)$	μ_{ij}	σ_{ij}	α_{ij}	d_{ij}	θ_{ij}	r_{ij}
11	0	1	1	0	0	0	q_{11}
12	11	0	0	$\pi/2$	0	q_{12}	0
13	12	0	2	0	$d_{13} = 0.0675\text{m}$	0	0
14	13	0	2	0	$d_{14} = 0.4505\text{m}$	0	0
15	14	0	2	0	$d_{15} = 0.082\text{m}$	0	0
16	15	0	2	0	$d_{16} = 0.065\text{m}$	0	0
17	16	0	2	0	$d_{17} = 0.14\text{m}$	0	0
21	0	1	1	0	$d_{21} = 0.4\text{m}$	0	q_{21}
22	21	0	0	$\pi/2$	0	q_{22}	0
23	22	0	0	0	$d_{10} = 0.0675\text{m}$	0	0
24	23	0	0	0	$d_{11} = 0.4505\text{m}$	0	0
25	24	0	0	0	$d_{12} = 0.082\text{m}$	q_{25}	0

Table 2: Rigid dynamic parameters for the links of the DualEMPS robot (SI units).

ij	m_{ij}	mx_{ij}	zz_{ij}	fs_{ij}	fv_{ij}
11	95.196	0.000	0.000	25	250
12	0.843	0.004	$7.06e - 4$	0.02	0.08
13	0.252	0.057	0.017	0.00	0.00
14	0.396	0.030	0.003	0.00	0.00
15	0.000	0.000	0.000	0.00	0.00
16	0.078	0.005	$5.12e - 4$	0.00	0.00
17	0.177	0.006	$2.56e - 4$	0.00	0.00
21	95.196	0.000	0.000	25	250
22	0.843	0.004	$7.06e - 4$	0.02	0.08
23	0.252	0.057	0.017	0.00	0.00
24	0.190	0.011	$9.44e - 4$	0.00	0.00
25	0.000	0.000	0.000	0.01	0.005

Table 3: Comparison of natural frequencies of the DualEMPS robot computed with the proposed model, computed with Adams and experimentally measured for $q_{11} = q_{21} = 0$.

(Hz)	f_1	f_2	f_3	f_4	f_5	f_6
Adams	14.41	24.92	49.78	97.91	110.86	219.60
Model	14.22	25.09	48.22	92.37	109.08	423.39
Measured ($\pm 1\text{Hz}$)	14.00	25.00	48.00	93.00	108.00	163.00

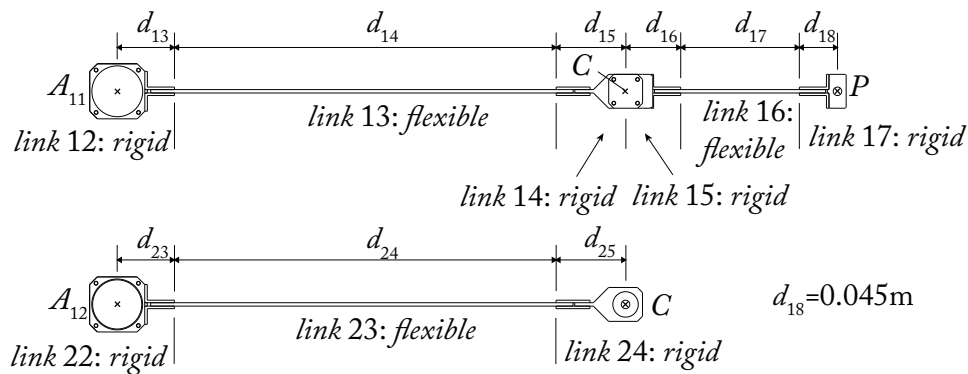


Figure 4: Top view of the CAD drawings of the two robot legs.

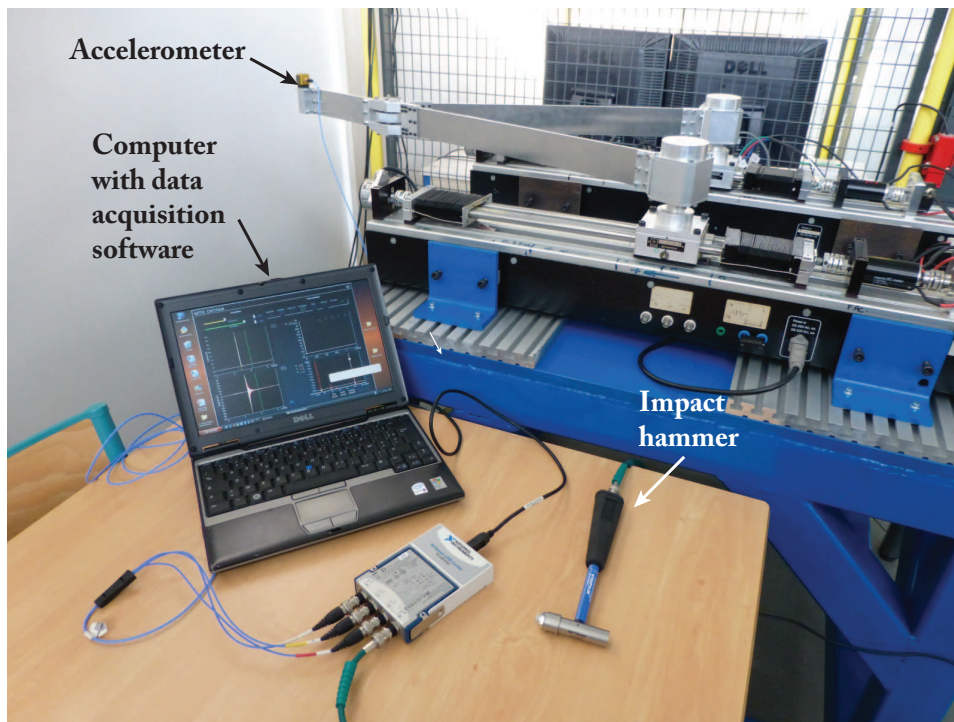


Figure 5: Experimental setup for measuring the natural frequencies

methods and then included in a S-function solved using Matlab/Simulink. The model includes 1041 intermediate variables and 1287 '+' or '-' and 1555 '*' or '/' operators. To the best of our knowledge, there exist no works that try to minimize the number of operators in the elastodynamic models of parallel robots. Therefore, the efficiency of the proposed solution may be difficult to analyze. However, for reasons of comparison, an Adams model interfaced with Simulink via the use of the module Adams/Control is also

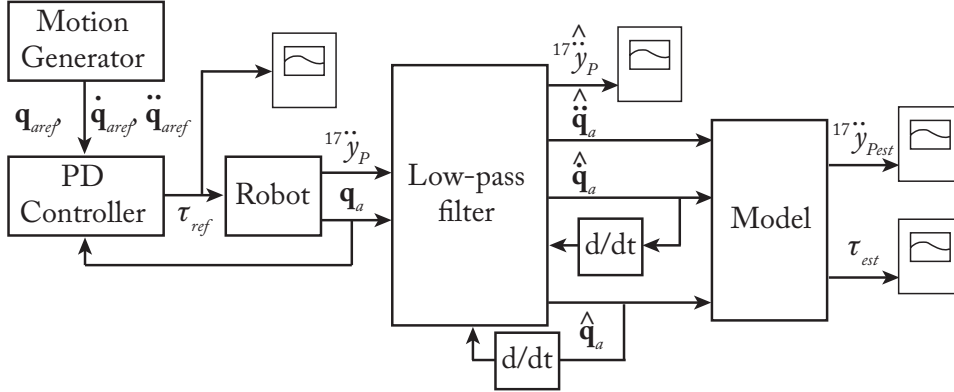


Figure 6: Measured and estimated data for the second experimental tests.

created. In this model, the elastic links are modeled using discrete flexible elements (after several tests on the Adams model accuracy, it is decided to use 8 discrete flexible element by elastic link).

Some first experimental tests are carried out for measuring the natural frequencies of the DualEMPS using the setup presented in Fig. 5. The application of experimental modal testing to the DualEMPS is done through impact hammer excitation, a 3-D accelerometer response and data post-processing, conducted using the DataBox software developed at IRCCyN and commercialized by MITIS company [MIT]. The impact point is near point P and the directions of excitation are contained in the horizontal plane in order to get the resonance frequencies that involves planar displacements of the robot. Piezoelectric triaxial accelerometers with a sensitivity of 1000 mV/g are used to measure the three acceleration responses. The acquisitions are performed for several robot configurations. However, the natural frequencies of the DualEMPS are near constant anywhere in the workspace, and all tested configurations lead to almost the same results. Therefore, only the results for the nominal configuration $q_{11} = q_{21} = 0$ will be presented. Each measurement resolution is equal to 1 Hz as the acquisition time and the sampling time are equal to 1 s and 40 μ s, respectively.

The resonance frequencies are obtained with a fast Fourier transform of the signals given by the triaxial accelerometer. As a result, the measured resonance frequencies between 0 and 200 Hz are given in Table 3. It is noteworthy that the resonance frequencies of the DualEMPS amount to its natural frequencies as the damping is considered negligible. The obtained results show that the five first frequencies predicted with our model are very close to the measured ones. However, as the model is made of three beam elements only, the frequency prediction after the fifth mode is not correct anymore. This prediction could be improved by introducing a higher number of elements, but this will increase the computational time. Nevertheless,

having a good prediction for all frequencies in the interval $[14, 110]$ Hz is already a very good result. Note that the Adams model prediction gives almost the same result as our model for the five first frequencies and is not correct for the sixth frequency (even if it is closer to reality than our model).

The second experimental tests are described thereafter. Some reference trajectories are introduced in the robot controller. During the robot displacements, three types of data are recorded (Fig. 6):

- the value of the input torques reference τ_{ref} given by the controller to the actuators;
- the actuator displacements \mathbf{q}_a given by the robot encoders;
- the acceleration ${}^{17}\ddot{y}_P$ of the point P (in the local \mathbf{y} direction) measured via the use of a uniaxial accelerometer with a sensitivity of 995mV/g.

Then, to make sure that the models give a correct estimation of the robot input torques and elastic displacements, the real measured actuator displacements are given as the model inputs, as well as the computed actuator velocities and accelerations, estimated from differentiation and low-pass filtering of the encoders data (Fig. 6). Results in terms of end-effector accelerations ${}^{17}\ddot{y}_{P_{est}}$ and computed input torques τ_{est} are finally compared with reality.

The first reference trajectory is a square in the joint space (Fig. 7(a)). The resulting displacement of the end-effector is a path following the workspace boundaries (Fig. 8). On each square edge, the velocity profile represented in Fig. 7(b) is applied: the actuator moves during 1.8s and then stops so that the free vibration of the robot can be observed. The measured acceleration ${}^{17}\ddot{y}_P$ of point P , low-pass filtered at 100Hz in order to suppress high-frequency noise, is plotted at Fig. 9 and compared with the acceleration computed with our model and with Adams. There are very good correlations between the measurements and the simulations (with better results for our model), even if the damping is higher in the simulations which can be due both to solver problems and to too large model approximation for the dissipative terms. For the simulations, Adams gives the result after several hours of computation while our model send the results in 6'05 min (for a Pentium 4 2.70GHz, 8Go of RAM).

The motor torques prediction (in prismatic joint side) is also compared with the measured actuator torques¹ in Figs. 10 and 11. The torque prediction is better for our model even if there are some problems of noise when the actuator velocity is very low (mainly due to problem of Coulomb friction modeling with the '*sign*' function).

¹For the Adams model, simulations are run without taking into account the friction effects on the actuated joints which are added by hand after the computation

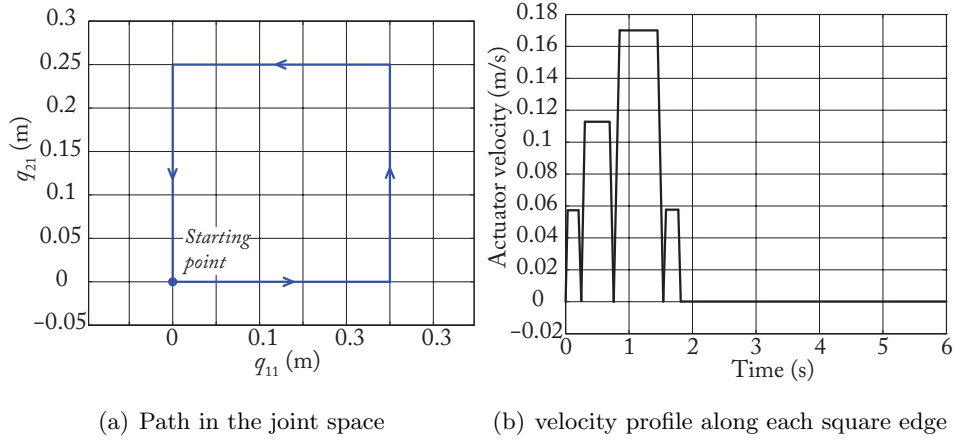


Figure 7: The first reference trajectory.

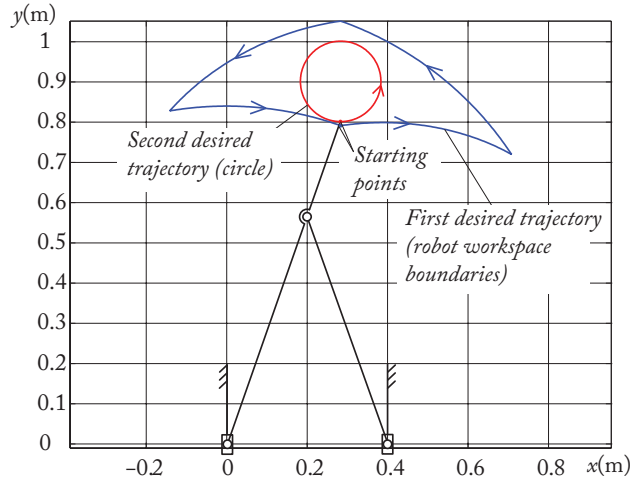
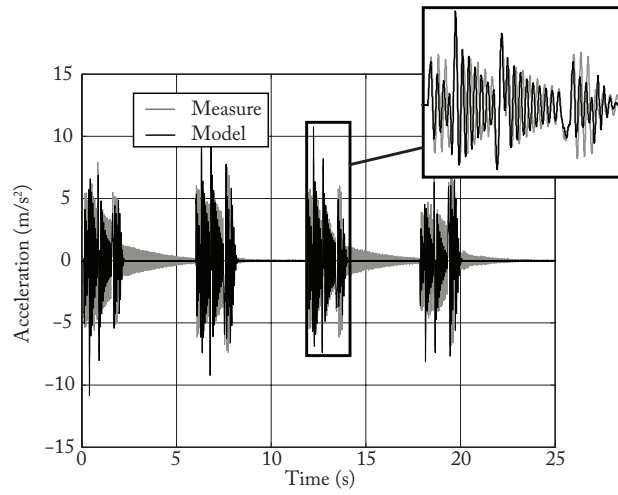


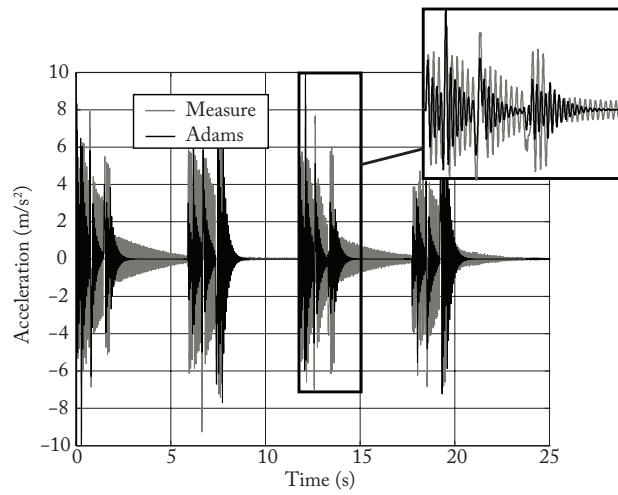
Figure 8: The reference trajectories in the workspace (scaled).

A second reference trajectory is introduced in the robot controller. This is a circle of radius 0.1m centered in [0.28, 0.90]m along which is applied a constant velocity profile of 0.20m/s (Fig. 8). The predictions of the acceleration ${}^{17}\ddot{y}_P$ of point P and of the input torques using our model are shown in Figs. 12, 13 and 14. There is still a very good correlation between the measurements and the simulations. Note that for this second simulation, data provided by Adams are not presented as we have just shown that Adams gives similar results and requires much longer computational time.

All these results show the validity and efficiency of our modeling approach.

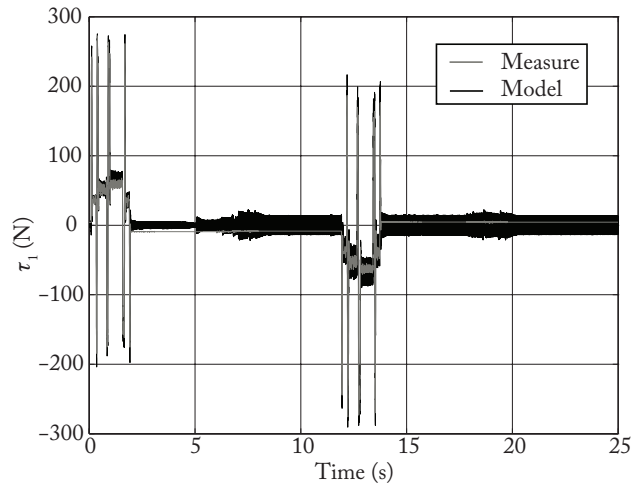


(a) Comparison of the measure and our model

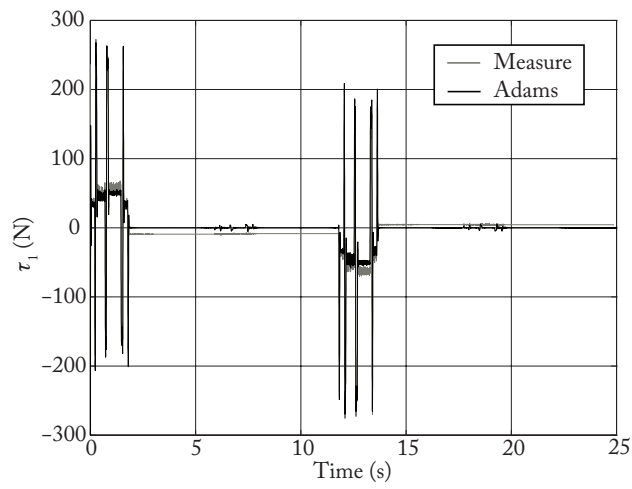


(b) Comparison of the measure and Adams

Figure 9: Point P acceleration for the first reference trajectory.

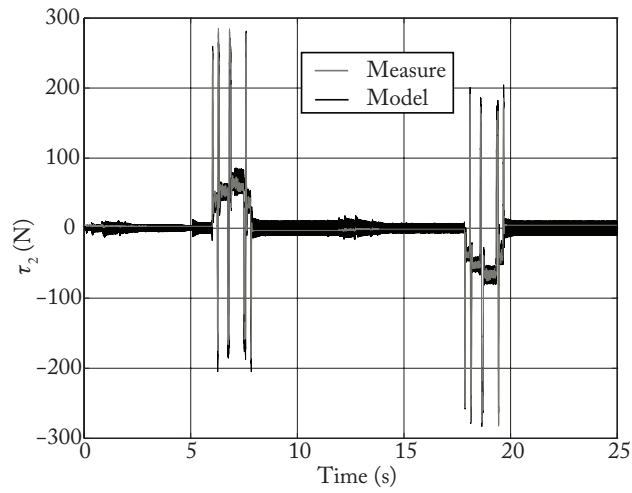


(a) Comparison of the measure and our model

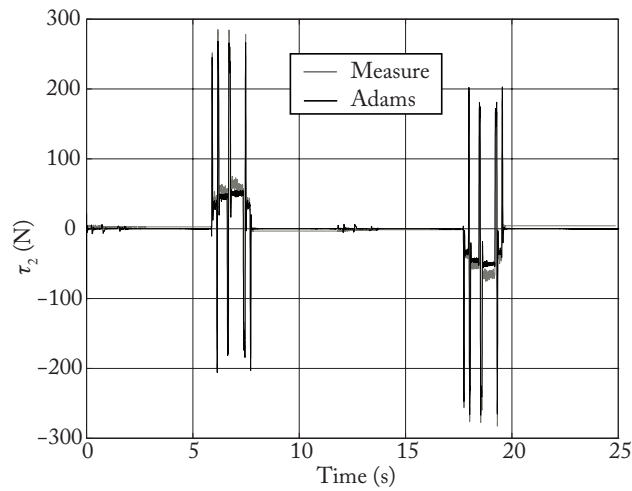


(b) Comparison of the measure and Adams

Figure 10: Input torque of the first actuator for the first reference trajectory.



(a) Comparison of the measure and our model



(b) Comparison of the measure and Adams

Figure 11: Input torque of the second actuator for the first reference trajectory.

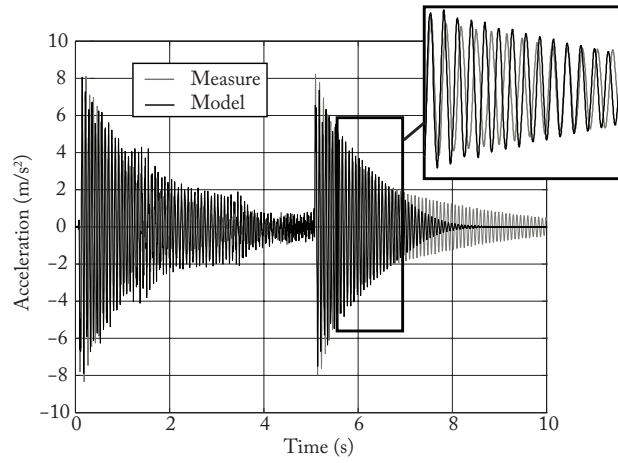


Figure 12: Comparison of the measured and computed acceleration of point P for the second reference trajectory.

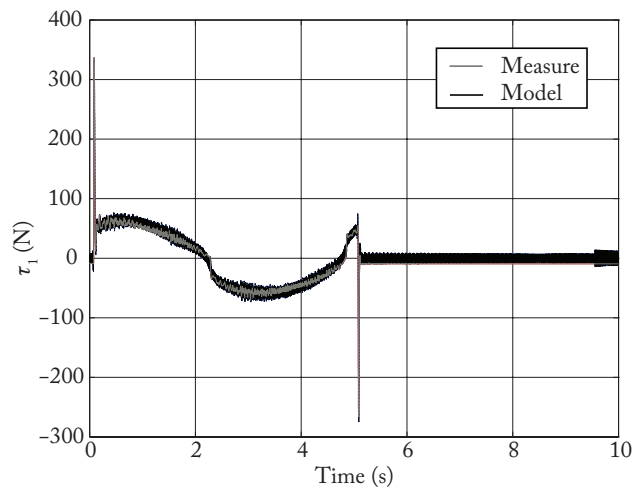


Figure 13: Comparison of the measured and computed actuator 1 input torque for the second reference trajectory.

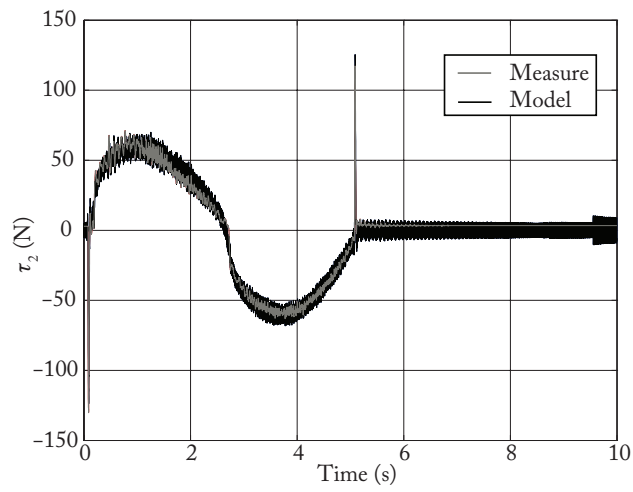


Figure 14: Comparison of the measured and computed actuator 2 input torque for the second reference trajectory.

6 Conclusions

This paper has presented a symbolic and recursive calculation of the dynamic model of flexible parallel robots. Such models are useful for several different reasons:

- In design optimization processes, optimization algorithms that test thousands of robot parameters are used. If the computational time required for the calculation of one iteration of the the elastodynamic model of the robot is not minimized, several days, and even month, can pass before the results are obtained.
- Symbolic expressions, with a minimized number of variables and operators, are requested for computing the identification model, in order to decrease the risk of error propagation due to the noisy measured data.

In order to reduce the computational time, the number of operators during the symbolic calculation of the model has been minimized. In order to achieve this goal, the Newton-Euler principle was used and combined with the principle of virtual powers. The Jacobian matrices defining the kinematic constraints have been computed using some recursions that decrease the number of operators. Using such a method, both link and joint flexibilities can be taken into account.

The proposed algorithm was used to compute the elastodynamic model of a prototype of a planar parallel robot developed at IRCCyN: the DualEMPS. The obtained model has been compared with a model created in Adams and with experimental measurements. It has been shown that our model gives correct estimations of the robot natural frequencies, end-effector displacements and input torques with a computational time largely inferior to the time required by Adams. All the obtained results have shown the validity of our approach.

It should be finally mentioned that our model computes the solution for a trajectory of 25 s in 6 minutes. This is due to the fact that, for estimating the elastic deformations, velocities and accelerations, a non linear differential equation must be solved. However, for real time control, if the values of the deformations are fed into the model (through the measurements of the deformations via accelerometers or any other sensor), the computational time is compatible with real time applications.

Future works will concern the development of methodologies for the identification of the elastodynamic models of parallel robots and the decrease of the required computational time by the use of model reduction techniques.

Appendix

Computation of the Elastodynamic Model of the Flexible Free Body Using the PVP

In [Boyer et al., 2007], the *PVP* is used for computing the elastodynamic model of a free flexible body. This principle can be written as:

$$P_{acc}^* = P_{int}^* + P_{ext}^* \quad (45)$$

where P_{acc}^* is the virtual power of due to the accelerations quantities, P_{int}^* is the virtual power due to the internal elastic efforts and P_{ext}^* is the virtual power of the external efforts. It is to be noticed that, in what follows in the paper, the superscript '*' stands for a virtual quantity.

By definition, the virtual power of due to the accelerations quantities is equal to [Boyer et al., 2007],

$$P_{acc}^* = \int_{\Sigma_j} \gamma_j^T(M_j) \mathbf{v}_j^*(M_j) dm \quad (46)$$

where Σ_j is the body under consideration, dm a small quantity of mass and [Boyer et al., 2007]

$$\begin{aligned} \gamma_j(M_j) = & \gamma_j(A_j) + \Phi_{d_j} \ddot{\mathbf{q}}_{e_j} + 2\omega_j(A_j) \times \Phi_{d_j} \dot{\mathbf{q}}_{e_j} \\ & + \omega_j(A_j) \times (\omega_j(A_j) \times \mathbf{r}_j(M_j)) + \alpha_j(A_j) \times \mathbf{r}_j(M_j) \end{aligned} \quad (47)$$

is the translational acceleration of the point M_j with respect to the base frame, with $\alpha_j(A_j)$ the rotational acceleration of the considered body at point A_j with respect to the base frame. Introducing (2) into (46), it comes that:

$$\begin{aligned} P_{acc}^* = & \left(\int_{\Sigma_j} \gamma_j(M_j) dm \right)^T \mathbf{v}_j^*(A_j) + \\ & \left(\int_{\Sigma_j} \mathbf{r}_j(M_j) \times \gamma_j(M_j) dm \right)^T \omega_j^*(A_j) + \\ & \left(\int_{\Sigma_j} \gamma_j^T(M_j) \Phi_{d_j}(M_{0j}) dm \right) \dot{\mathbf{q}}_{e_j}^* \end{aligned} \quad (48)$$

The virtual power of the external efforts can be decomposed into two parts

$$P_{ext}^* = P_{grav}^* + P_{reac}^* \quad (49)$$

where $P_{grav}^* = \int_{\Sigma_j} \left(\mathbf{v}_j^*(M_j) \right)^T \mathbf{g} dm$ is the virtual power of the gravity field \mathbf{g} and $P_{reac}^* = \mathbf{f}_{A_j}^T \mathbf{v}_j^*(A_j) + \mathbf{m}_{A_j}^T \omega_j^*(A_j) - \mathbf{f}_{B_j}^T \mathbf{v}_j^*(B_j) - \mathbf{m}_{B_j}^T \omega_j^*(B_j)$ the virtual

power due to the reactions at point A_j and B_j (Fig. 2(a)), where \mathbf{f}_{A_j} (\mathbf{f}_{B_j} , resp.) is the force applied at point A_j (B_j , resp.) and \mathbf{m}_{A_j} (\mathbf{m}_{B_j} , resp.) is the moment applied at point A_j (B_j , resp.). Thus, introducing (2) in the previous expressions,

$$\begin{aligned}
P_{grav}^* &= \left(\int_{\Sigma_j} \mathbf{g} \, dm \right)^T \mathbf{v}_j^*(A_j) + \\
&\quad \left(\int_{\Sigma_j} \mathbf{r}_j(M_j) \times \mathbf{g} \, dm \right)^T \omega_j^*(A_j) + \\
&\quad \left(\int_{\Sigma_j} \mathbf{g}^T \Phi_{d_j}(M_{0j}) dm \right) \dot{\mathbf{q}}_{e_j}^*
\end{aligned} \tag{50}$$

and

$$\begin{aligned}
P_{reac}^* &= (\mathbf{f}_{A_j} - \mathbf{f}_{B_j})^T \mathbf{v}_j^*(A_j) + \\
&\quad (\mathbf{m}_{A_j} - \mathbf{m}_{B_j} - \mathbf{r}_j(B_j) \times \mathbf{f}_{B_j})^T \omega_j^*(A_j) - \\
&\quad \left(\mathbf{f}_{B_j}^T \Phi_{d_j}(B_{0j}) + \mathbf{m}_{B_j}^T \Phi_{r_j}(B_{0j}) \right) \dot{\mathbf{q}}_{e_j}^*
\end{aligned} \tag{51}$$

Finally, the virtual power due to the internal elastic efforts can be written as [Boyer et al., 2007]:

$$P_{int}^* = -\mathbf{q}_{e_j} \mathbf{K}_{ee_j} \dot{\mathbf{q}}_{e_j}^* \tag{52}$$

where \mathbf{K}_{ee_j} is the stiffness matrix of the considered body that can be computed using the expressions presented in [Shabana, 2005].

Thus, introducing (47), (48), (50), (51) and (52) into (45), developing and simplifying the expressions, and taking into account the fact that the virtual velocities $\mathbf{v}_j^*(A_j)$, $\omega_j^*(A_j)$ and $\dot{\mathbf{q}}_{e_j}^*$ are independant, three sets of equilibrium equations can be obtained:

$$\begin{aligned}
\Delta \mathbf{f}_{c_j} &= m_j (\gamma_j(A_j) - \mathbf{g}) + \mathbf{M} \mathbf{S}_{de_j} \ddot{\mathbf{q}}_{e_j} \\
&\quad - \mathbf{M} \mathbf{S}_j \times \alpha_j(A_j) + \omega_j(A_j) \times (\omega_j(A_j) \times \mathbf{M} \mathbf{S}_j) \\
&\quad + 2\omega_j(A_j) \times \mathbf{M} \mathbf{S}_{de_j} \dot{\mathbf{q}}_{e_j}
\end{aligned} \tag{53}$$

$$\begin{aligned}
\Delta \mathbf{m}_{c_j} &= \mathbf{M} \mathbf{S}_j \times (\gamma_j(A_j) - \mathbf{g}) + \mathbf{M} \mathbf{S}_{re_j} \ddot{\mathbf{q}}_{e_j} \\
&\quad + \omega_j(A_j) \times (\mathbf{I}_j \omega_j(A_j)) + 2 \sum_{k=1}^{N_j} \mathbf{I}_{re_{kj}} \omega_j(A_j) \dot{q}_{ek_j} \\
&\quad + 2 \sum_{i,k=1}^{N_j} \mathbf{I}_{ee_{ik_j}} \omega_j(A_j) q_{ei_j} \dot{q}_{ek_j} + \mathbf{I}_j \alpha_j(A_j)
\end{aligned} \tag{54}$$

and

$$\begin{aligned}
\Delta \mathbf{s}_{c_j}|_i &= \mathbf{MS}_{de_j}^T|_i (\gamma_j(A_j) - \mathbf{g}) \\
&+ \mathbf{MS}_{re_j}^T|_i \alpha_j(A_j) + \mathbf{M}_{ee_j}|_i \ddot{\mathbf{q}}_{e_j} \\
&+ 2 \sum_{k=1}^{N_j} \lambda_{ki} \omega_j(A_j) \dot{q}_{ekj} - \omega_j(A_j)^T \mathbf{I}_{re_{ij}}^T \omega_j(A_j) \\
&- \sum_{k=1}^{N_j} \omega_j(A_j)^T \mathbf{I}_{ee_{ikj}} \omega_j(A_j) q_{ekj} + \mathbf{K}_{ee_j}|_i \mathbf{q}_{e_j}
\end{aligned} \tag{55}$$

where the symbol ' $|_i$ ' denotes the line i of the considered vector or matrix, m_j is the mass of the body j and the terms \mathbf{MS}_j , \mathbf{MS}_{de_j} , \mathbf{MS}_{re_j} , \mathbf{I}_j , $\mathbf{I}_{re_{ij}}$, $\mathbf{I}_{ee_{ikj}}$, $\Delta \mathbf{f}_{c_j}$, $\Delta \mathbf{m}_{c_j}$ and $\Delta \mathbf{s}_{c_j}$ are defined in the following expressions:

$$\begin{aligned}
\mathbf{MS}_j &= \int_{\Sigma_j} \mathbf{r}_j(M_j) dm \\
&= \int_{\Sigma_j} \mathbf{r}_j(M_{0j}) dm + \int_{\Sigma_j} \Phi_{d_j}(M_{0j}) dm \mathbf{q}_{e_j} \\
&= \mathbf{MS}_{r_j} + \mathbf{MS}_{de_j} \mathbf{q}_{e_j}
\end{aligned} \tag{56}$$

is the global vector of the first moments of inertia (\mathbf{MS}_{r_j} is the 3×1 constant vector of the first moments of inertia of the rigid link and \mathbf{MS}_{de_j} a $3 \times N_j$ constant matrix),

$$\begin{aligned}
\mathbf{MS}_{re_j} &= \int_{\Sigma_j} \hat{\mathbf{r}}_j(M_j) \Phi_{d_j}(M_{0j}) dm \\
&= \int_{\Sigma_j} \hat{\mathbf{r}}_j(M_{0j}) \Phi_{d_j} dm \\
&+ \int_{\Sigma_j} \Phi_{d_j}(M_{0j}) \mathbf{q}_{e_j} \times \Phi_{d_j}(M_{0j}) dm \\
&= \sum_{k=1}^{N_j} \left(\beta_k + \sum_{i=1}^{N_j} \lambda_{ki} \mathbf{q}_{ei_j} \right)
\end{aligned} \tag{57}$$

with $\hat{\mathbf{r}}_j$ the cross-product matrix associated with the vector \mathbf{r}_j and

$$\beta_k = \int_{\Sigma_j} \mathbf{r}_j(M_{0j}) \times \phi_{dk_j}(M_{0j}) dm \tag{58}$$

and

$$\lambda_{ki} = \int_{\Sigma_j} \phi_{dk_j}(M_{0j}) \times \phi_{di_j}(M_{0j}) dm \tag{59}$$

two 3×1 constant vectors,

$$\begin{aligned}
\mathbf{I}_j &= \int_{\Sigma_j} \hat{\mathbf{r}}_j^T(M_j) \hat{\mathbf{r}}_j(M_j) dm = \int_{\Sigma_j} \hat{\mathbf{r}}_j^T(M_{0j}) \hat{\mathbf{r}}_j(M_{0j}) dm \\
&+ \int_{\Sigma_j} \sum_{k=1}^{N_j} \hat{\mathbf{r}}_j^T(M_{0j}) \hat{\phi}_{dk_j}(M_{0j}) q_{ek_j} dm \\
&+ \int_{\Sigma_j} \sum_{k=1}^{N_j} \hat{\phi}_{dk_j}^T(M_{0j}) \hat{\mathbf{r}}_j(M_{0j}) q_{ek_j} dm \\
&+ \int_{\Sigma_j} \sum_{i,k=1}^{N_j} \hat{\phi}_{di_j}^T(M_{0j}) \hat{\phi}_{dk_j}(M_{0j}) q_{ei_j} q_{ek_j} dm \\
&= \mathbf{I}_{rr_j} + \sum_{k=1}^{N_j} \left(\mathbf{I}_{re_{kj}} + \mathbf{I}_{re_{kj}}^T \right) q_{ek_j} + \sum_{i,k=1}^{N_j} \mathbf{I}_{ee_{ik_j}} q_{ei_j} q_{ek_j}
\end{aligned} \tag{60}$$

is the global matrix of the second moments of inertia (\mathbf{I}_{rr_j} is the 3×3 constant matrix of the second moments of inertia of the rigid link and $\mathbf{I}_{re_{kj}}$ and $\mathbf{I}_{ee_{ik_j}}$ 3×3 constant matrices), in which $\hat{\phi}_{di_j}$ is the cross-product matrix associated with the vector ϕ_{di_j} ,

$$\Delta \mathbf{f}_{c_j} = \mathbf{f}_{A_j} - \mathbf{f}_{B_j} \tag{61}$$

$$\Delta \mathbf{m}_{c_j} = \mathbf{m}_{A_j} - \mathbf{m}_{B_j} - \mathbf{r}_j(B_j) \times \mathbf{f}_{B_j} \tag{62}$$

and

$$\Delta \mathbf{s}_{c_j} = \mathbf{f}_{B_j}^T \Phi_{d_j}(B_{0j}) + \mathbf{m}_{B_j}^T \Phi_{r_j}(B_{0j}) \tag{63}$$

Finally, the generalized *NE* model of a flexible free body can be written under the following matrix form:

$$\begin{aligned}
\begin{bmatrix} \Delta \mathbf{f}_{c_j} \\ \Delta \mathbf{c}_{c_j} \\ \Delta \mathbf{s}_{c_j} \end{bmatrix} &= \begin{bmatrix} m_j \mathbf{I}_{d3} & \widehat{\mathbf{M}} \mathbf{S}_j^T & \mathbf{M} \mathbf{S}_{de_j} \\ \widehat{\mathbf{M}} \mathbf{S}_j & \mathbf{I}_j & \mathbf{M} \mathbf{S}_{re_j} \\ \mathbf{M} \mathbf{S}_{de_j}^T & \mathbf{M} \mathbf{S}_{re_j}^T & \mathbf{M}_{ee_j} \end{bmatrix} \begin{bmatrix} \gamma_j(A_j) \\ \alpha_j(A_j) \\ \ddot{\mathbf{q}}_{e_j} \end{bmatrix} + \\
\begin{bmatrix} \mathbf{f}_{in_j} \\ \mathbf{c}_{in_j} \\ \mathbf{s}_{in_j} \end{bmatrix} &+ \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{K}_{ee_j} \mathbf{q}_{e_j} \end{bmatrix} + \begin{bmatrix} \mathbf{f}_{g_j} \\ \mathbf{c}_{g_j} \\ \mathbf{s}_{g_j} \end{bmatrix} = \mathbf{M}_j \begin{bmatrix} \dot{\mathbf{t}}_j \\ \ddot{\mathbf{q}}_{e_j} \end{bmatrix} + \mathbf{c}_j
\end{aligned} \tag{64}$$

where

$$\mathbf{f}_{in_j} = \omega_j(A_j) \times \left(\widehat{\mathbf{M}} \mathbf{S}_j^T \omega_j(A_j) + 2 \mathbf{M} \mathbf{S}_{de_j} \dot{\mathbf{q}}_{e_j} \right) \tag{65}$$

$$\begin{aligned}
\mathbf{c}_{in_j} &= \omega_j(A_j) \times \left(\mathbf{I}_j \omega_j(A_j) \right) + 2 \sum_{k=1}^{N_j} \mathbf{I}_{re_{kj}} \omega_j(A_j) \dot{q}_{ek_j} \\
&+ 2 \sum_{i,k=1}^{N_j} \mathbf{I}_{ee_{ik_j}} \omega_j(A_j) q_{ei_j} \dot{q}_{ek_j}
\end{aligned} \tag{66}$$

$$\mathbf{s}_{in_j|i} = 2 \sum_{k=1}^{N_j} \lambda_{ki} \omega_j(A_j) \dot{q}_{ek_j} - \omega_j(A_j)^T \mathbf{I}_{re_{ij}}^T \omega_j(A_j) - \sum_{k=1}^{N_j} \omega_j(A_j)^T \mathbf{I}_{e_{ik_j}} \omega_j(A_j) q_{ek_j} \quad (67)$$

$$\mathbf{f}_{g_j} = -m_j \mathbf{g}, \mathbf{c}_{g_j} = -\widehat{\mathbf{M}} \mathbf{S}_j \mathbf{g}, \mathbf{s}_{g_j} = -\mathbf{M} \mathbf{S}_{de_j}^T \mathbf{g} \quad (68)$$

Acknowledgements

This work was supported by the French ANR ARROW (ANR 2011BS3 006 01).

The authors would like to thank Maxime Gautier and Anthony Jubien for their precious help with the prototype and during the experiments.

References

<http://www.mitis.fr/>. URL <http://www.mitis.fr/>.

O.A. Bauchau. *Flexible multibody dynamics*. Springer, 2011.

R.D. Blevins. *Formulas for natural frequency and mode shape*. Krieger Publishing Company, 2001.

B. C. Bouzgarrou, P. Ray, and G. Gogu. New approach for dynamic modelling of flexible manipulators. *Part K: Journal of Multi-body Dynamics*, 219(3):285–298, 2005.

F. Boyer and W. Khalil. An efficient calculation of the flexible manipulator inverse dynamics. *International Journal of Robotics Research*, 17(3):282–293, 1998.

F. Boyer, W. Khalil, M. Benosman, and G. LeVey. *Robot Manipulators. Modeling, Performance, Analysis and Control*, chapter 7: Modeling and Control of Flexible Robots, pages 337–394. Control Systems, Robotics and Manufacturing Series. ISTE, 2007.

S. Briot and W. Khalil. Recursive symbolic calculation of the dynamic model of flexible parallel robots. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013)*, Karlsruhe, Germany, May 6-10 2013.

S. Briot, A. Pashkevich, and D. Chablat. Reduced elastodynamic modelling of parallel robots for the computation of their natural frequencies. In *World Congress in Mechanism and Machine Science*, 13, June 2011.

- R.R. Craig. *Structural dynamics*. Wiley, 1981.
- R.R. Craig and M.C.C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6(7), 1968.
- J.G. de Jalon and E. Bayo. *Kinematic and dynamic simulations of multibody systems*. Springer-Verlag.
- S.K. Dwivedy and P. Eberhard. Dynamic analysis of flexible manipulators, a literature review. *Mechanism and Machine Theory*, 41(7):749–777, 2006.
- O. Ibrahim and W. Khalil. Inverse and direct dynamic models of hybrid robots. *Mechanism and Machine Theory*, 45:627–640, 2010.
- W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton London, 2002.
- W. Khalil and M. Gautier. Modeling of mechanical systems with lumped elasticity. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3965–3970, San Francisco, CA, USA, 2000.
- W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. *Journal of Intelligent and Robotic Systems*, 49(1):19–37, 2007.
- J. Kruszewski, W. Gawronski, E. Wittbrodt, F. Najbar, and S. Grabowski. *The rigid finite element method*. Arkady, Warszawa, 1975.
- F.C. Moon. *Applied dynamics*. J. Wiley and Sons, 2007.
- M. Rognant, E. Courteille, and P. Maurine. A systematic procedure for the elastodynamic modeling and identification of robot manipulators. *IEEE Transactions on Robotics*, 26(6):1085–1093, 2010.
- A.A. Shabana. Dynamics of flexible bodies using generalized newton-euler equations. *Journal of Dynamic Systems, Measurement, and Control*, 112: 496–503, 1990.
- A.A. Shabana. *Dynamics of Multibody Systems*. Cambridge University Press, 2005.
- I. Sharf and C. Damaren. Simulation of flexible-link manipulators: basis functions and non-linear terms in the motion equations. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 1992)*, pages 1956–1962, Nice, France, 1992.
- K. Stachera and W. Schumacher. *Automation and Robotics*, chapter 15: Derivation and Calculation of the Dynamics of Elastic Parallel Manipulators. I. Tech. Educ. & Publishing, 2008.

E. Wittbrodt, I. Adamiec-Wójcik, and S. Wojciech. *Dynamics of Flexible Multibody Systems*. Springer-Verlag Berlin Heidelberg, 2006.