Recursive Symbolic Calculation of the Dynamic Model of Flexible Parallel Robots

Sébastien Briot[1] and Wisama Khalil[2]

Abstract— **This paper presents a symbolic and recursive calculation of the dynamic model of flexible parallel robots. In order to reduce the computational time, 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, but to the best of our knowledge, nothing has been developed for parallel robots with distributed flexibilities. This paper aims at filling this gap. In order to minimize the number of operations, the Newton-Euler principle is used and combined with the principle of virtual powers. The Jacobian matrices defining the kinematic constraints are computed using recursive calculations that decrease the number of operators. The proposed algorithm is used to compute the elastodynamic model of a planar parallel robot. The obtained results, compared with those obtained with commercial softwares, show the validity of the proposed algorithm.**

I. INTRODUCTION

The way to obtain the full dynamic model of rigid parallel robots has been deeply analyzed [1], [2], [3], however several open questions still arise for the computation of their elastodynamic model. One of them concerns the reduction of the computational time that is generally huge. To decrease the computational cost, one can either (i) decrease the number of variables (using model reduction methods [4], [5], [6] and truncated series of shape functions [7]) or (ii) optimize the symbolic computation of the model to minimize the number of operators (similarly to what has been done for rigid robots [8], robots with lumped springs [9] or for serial robots with distributed flexibilities [10]). It is obvious that both methods can also be combined. However, this paper only focuses on the way to optimize the symbolic computation of the elastodynamic model of parallel robots.

Two main approaches are generally used for computing the elastodynamic model (see [11] for a large literature review): (i) lumped modeling [9], [12], [13] and (ii) modeling using distributed flexibilities [10], [14], [15], [16], [17], [18]. The lumped modeling is generally simpler to apply but tends to decrease the accuracy of the model. In [9], 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 allow the minimization of the number of operators in the model. In [13], 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 larger computational time.

On the contrary, using distributed flexibilities allows the increasing of the model accuracy. But it requires higher skills. In [14], [15], [17], [18], the authors proposed some general methodologies based on the Lagrange principle that can be applied to any system. For taking into account the kinematic dependencies such as in the case of close-loop mechanisms, some Jacobian matrices are used. In [16], the authors combine the Lagrange principle and the principle of virtual works for derivating 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 allow the minimization of the number of operators for the symbolic computation of the model. To the best of our knowledge, a systematic procedure to compute the elastodynamic model (using distributed flexibilities) of parallel robot with a minimum numbers of operators has never been proposed.

The present work aims at filling this gap. In order to minimize the number of operations, the Newton Euler (NE) principle (which is known to reduce the number of operators [8], [19]) 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 method proposed in [1], [2] for rigid robots and in [16] for flexible robots is used. This method 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 plus (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.

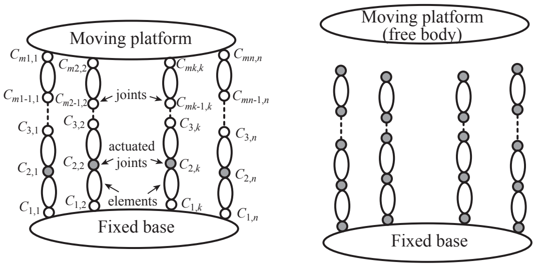
This method is effective, systematic and can be applied to any parallel robots.

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. Section 5 presents some illustrative example. Finally, in Section 6, conclusions are drawn.

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

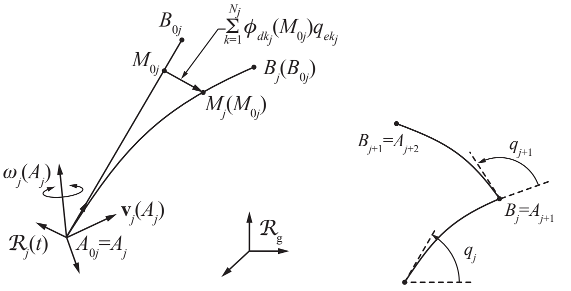
[1] S. Briot is with the French CNRS and the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), 44321 Nantes France (Sebastien.Briot@irccyn.ec-nantes.fr)

[2] W. Khalil is with the LUNAM, Ecole Centrale de Nantes and the IRCCyN, 44321 Nantes France (Wisama.Khalil@irccyn.ec-nantes.fr)



(a) Kinematic chain j (b) Virtual tree structure

Fig. 1. A general parallel robot.



(a) Parameters of one flexible link j (b) Assembly of two flexible links

Fig. 2. Schematics of the flexible elements into consideration.

II. RECALL ON THE GENERALIZED NEWTON-EULER MODEL OF A FLEXIBLE FREE BODY

This section aims at making some brief recall on the way to obtain the generalized NE model of a flexible free body. For further development, the reader is referred to [10], [19].

A. 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 position vector of 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 $\Phi_{d,r_j} = \left[\phi_{d,r1_j}, \cdots, \phi_{d,rN_{j_j}}\right]$, $\phi_{dk_j}(M_{0j})$ 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 $\dot{\mathbf{q}}_{e_j} = \left[\dot{q}_{e1_j}, \cdots, \dot{q}_{eN_{j_j}}\right]^T$, $\dot{q}_{ek_j}$ 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 $\mathbf{r}_j(M_{0_j})$ is the position of the point M_{0j} with respect to the local frame and $\mathbf{q}_{e_j} = \left[q_{e1_j}, \cdots, q_{eN_{j_j}}\right]^T$ 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 $3N_j$ supplementary variables ($\mathbf{q}_{e_j}$, $\dot{\mathbf{q}}_{e_j}$ and $\ddot{\mathbf{q}}_{e_j}$) and $6N_j$ shape functions compared with the rigid free body case.

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

B. Generalized Newton-Euler model of a flexible free body

Skipping all mathematical derivations and referring the reader to [10], the generalized NE model of a flexible free body can be obtained via the application of the PVP. This model is known to reduce the number of operators and takes the form:

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

where

- m_j the total mass of the body j
- $\mathbf{I_d}_k$ 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,
- $\dot{\mathbf{t}}_j = \left[\gamma_j(A_j)^T, \alpha_j(A_j)^T\right]^T$ is the acceleration screw 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,
- $\Delta\mathbf{f}_{c_j} = \mathbf{f}_{c_j} - \mathbf{f}_{c_{j+1}}$ and $\Delta\mathbf{c}_{c_j} = \mathbf{c}_{c_j} - \mathbf{c}_{c_{j+1}} - \mathbf{r}_j(A_{j+1}) \times \mathbf{f}_{c_{j+1}}$ are the total sum of forces and torques transmitted by the joints, $\mathbf{f}_{c_j}$ and $\mathbf{c}_{c_j}$ being the reaction forces and torques in joint j, respectively,
- $\Delta\mathbf{s}_{c_j} = -\Phi_{dj}^T(B_{0j})\mathbf{f}_{c_{j+1}}$ - $\Phi_{rj}^T(B_{0j})\mathbf{c}_{c_{j+1}}$ is the total sum of the elastic generalized forces transmitted by the joints.

For limiting the size of the paper, the expressions of these terms are not given here. However, the reader can refer to [10] for more details.

III. 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 - $i = 1, ..., n$) (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 degrees of freedom (DOF) of the parallel robot. The number of shape functions by element is denoted as N_{ij} ($j = 1, ..., m_i - 1$). 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 $\mathbf{q}_t = \left[\mathbf{q}_a^T, \mathbf{q}_p^T, \mathbf{q}_e^T\right]^T$.

A. Application of the principle of virtual powers

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 the symbol $'*'$ stands for a virtual velocity, $\tau_{t_{ij}}$ 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 lead 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.

B. 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 [8].

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) [10]:

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

which can also be written as:

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

with

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

where

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

In the matrix $\Phi_{\mathbf{q}_{eij}}$, the term ${}^{ij}\mathbf{R}_{i(j-1)}\Phi_{i(j-1)}(A_{ij})$ 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:

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

where ${}^{ij}\mathbf{Rot}_{i(j-1)}$ 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:

$$\mathbf{J}_{ij} = \begin{bmatrix} \mathbf{J}_{t_{ij}} \\ \mathbf{O}_{\mathbf{q}_{eij}} \end{bmatrix} \tag{17}$$

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

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

C. Recursive computation of the accelerations

Differentiating (10), it can be shown that [10]:

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

where the expression of ${}^{ij}\mathbf{h}_{ij}$ is given in [10].

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

$${}^{ij}\dot{\mathbf{t}}_{ij} = \mathbf{J}_{t_{ij}} \ddot{\mathbf{q}}_t + {}^{ij}\mathbf{g}_{ij} \tag{19}$$

with

$${}^{ij}\mathbf{g}_{ij} = {}^{ij}\mathbf{h}_{ij} + {}^{ij}\mathbf{T}_{i(j-1)} {}^{i(j-1)}\mathbf{p}_{i(j-1)} \tag{20}$$

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

$$\begin{bmatrix} {}^{ij}\dot{\mathbf{t}}_{ij} \\ \ddot{\mathbf{q}}_{e_{ij}} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{t_{ij}} \\ \mathbf{O}_{\mathbf{q}_{eij}} \end{bmatrix} \ddot{\mathbf{q}}_t + \begin{bmatrix} {}^{ij}\mathbf{g}_{ij} \\ \mathbf{0} \end{bmatrix} = \mathbf{J}_{ij} \ddot{\mathbf{q}}_t + {}^{ij}\mathbf{g}^s_{ij} \tag{21}$$

D. Elastodynamic model of the virtual system

Introducing (21) into (9) leads to:

$$\begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} = \sum_{i,j} \mathbf{J}^T_{ij} \mathbf{M}_{ij} \mathbf{J}_{ij} \ddot{\mathbf{q}}_t + \mathbf{c}^s_{ij} \tag{22}$$

where

$$\mathbf{c}^s_{ij} = \mathbf{J}^T_{ij} \left(\mathbf{c}_{ij} + \mathbf{M}_{ij} {}^{ij}\mathbf{g}^s_{ij} \right) \tag{23}$$

The NE equations for the rigid moving platform are given by [2]:

$$\mathbf{f}_p = \mathbf{M}_p \dot{\mathbf{t}}_p + \mathbf{c}_p \tag{24}$$

where $\mathbf{f}_p$ are the platform reaction forces 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:

$$\begin{split} \begin{bmatrix} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} \\ \mathbf{f}_p \end{bmatrix} = \begin{bmatrix} \sum_{i,j} \mathbf{J}^T_{ij} \mathbf{M}_{ij} \mathbf{J}_{ij} & \mathbf{M}_p \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \begin{bmatrix} \mathbf{c}^s_{ij} \\ \mathbf{c}_p \end{bmatrix} \\ = \mathbf{M}^s_t \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{c}^s_t \end{split} \tag{25}$$

Adding the contributions of the motor inertia and friction effects [8]:

$$\begin{split} \begin{bmatrix} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} \\ \mathbf{f}_p \end{bmatrix} = \mathbf{M}^s_t \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{c}^s_t + \begin{bmatrix} \mathbf{I}_t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \\ \begin{bmatrix} \mathbf{F}_v \dot{\mathbf{q}}_t \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{F}_s sign\left(\dot{\mathbf{q}}_t\right) \\ \mathbf{0} \end{bmatrix} = \mathbf{M}_t \begin{bmatrix} \ddot{\mathbf{q}}_t \\ \dot{\mathbf{t}}_p \end{bmatrix} + \mathbf{c}_t \end{split} \tag{26}$$

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

IV. COMPUTATION OF THE ELASTODYNAMIC MODEL OF PARALLEL ROBOTS

To compute the elastodynamic model of the actual parallel robot, the loops must be closed by using the PVP. This is developed in the next section.

A. Application of the principle of virtual powers

Considering the actual robot, the PVP states that:

$$\begin{bmatrix} \dot{\mathbf{q}}^{T*}_t & \mathbf{t}^{T*}_p \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} \\ \mathbf{f}_p \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}}^{T*}_a & \dot{\mathbf{q}}^{T*}_e \end{bmatrix} \begin{bmatrix} \tau \\ \mathbf{0}_{n_e} \end{bmatrix} \tag{27}$$

where $\mathbf{t}_p$ is the platform twist and τ the actual actuator input efforts.

The velocities $\mathbf{t}_p$ and $\dot{\mathbf{q}}_t$ are linked to the actual generalized velocities $\dot{\mathbf{q}} = \left[\dot{\mathbf{q}}^T_a, \dot{\mathbf{q}}^T_e\right]^T$ by the relation:

$$\begin{bmatrix} \dot{\mathbf{q}}^T_t & \mathbf{t}^T_p \end{bmatrix}^T = \mathbf{J}\dot{\mathbf{q}} \tag{28}$$

where $\mathbf{J}$ is the global Jacobian matrix of the robot whose expression will be given in the following section.

Introducing (28) into (27) leads to

$$\begin{bmatrix} \dot{\mathbf{q}}^{T*}_a & \dot{\mathbf{q}}^{T*}_e \end{bmatrix} \mathbf{J}^T \begin{bmatrix} \begin{bmatrix} \tau_t \\ \mathbf{0}_{n_e} \end{bmatrix} \\ \mathbf{f}_p \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}}^{T*}_a & \dot{\mathbf{q}}^{T*}_e \end{bmatrix} \begin{bmatrix} \tau \\ \mathbf{0}_{n_e} \end{bmatrix} \tag{29}$$

or also, for any $\dot{\mathbf{q}}^*_a$ and $\dot{\mathbf{q}}^*_e$,

$$\begin{bmatrix} \tau \\ \mathbf{0}_{n_e} \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} \tau^T_t & \mathbf{0}^T_{n_e} & \mathbf{f}^T_p \end{bmatrix}^T \tag{30}$$

B. Computation of the passive joints and platform velocities and of the global Jacobian matrix

The global Jacobian matrix can be computed using the following method:

1) Express, in the based frame, the twist ${}^0\mathbf{t}_{im_i}$ of the virtual end-effector of each serial kinematic chain (points $C_{m_i,i}$ in Fig. 1(a)) as a function of $\dot{\mathbf{q}}_t$ using (12-14); ${}^0\mathbf{t}_{im_i} = \mathbf{J}_{t_i}\dot{\mathbf{q}}_t$
2) Express, in the based frame, the twist ${}^0\mathbf{t}_{im_i}$ of the virtual end-effector of each serial kinematic chain as a function of $\mathbf{t}_p$; ${}^0\mathbf{t}_{im_i} = \mathbf{J}_{p_i}\mathbf{t}_p$
3) Then assemble the two expressions and rearrange it in order to obtain the global Jacobian matrix $\mathbf{J}$ such as:

$$\begin{bmatrix}{}^0\mathbf{t}_{1m_1}\\ \vdots \\ {}^0\mathbf{t}_{nm_n}\end{bmatrix} = \mathbf{J}_p\mathbf{t}_p = \mathbf{J}_t\dot{\mathbf{q}}_t = \begin{bmatrix}\mathbf{J}_{t_a} & \mathbf{J}_{t_p} & \mathbf{J}_{t_e}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_p\\ \dot{\mathbf{q}}_e\end{bmatrix} \qquad (31)$$

If the robot is isostatic, (31) can be rewritten as:

$$\begin{bmatrix}-\mathbf{J}_{t_p} & \mathbf{J}_p\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_p\\ \mathbf{t}_p\end{bmatrix} = \begin{bmatrix}\mathbf{J}_{t_a} & \mathbf{J}_{t_e}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_e\end{bmatrix} \qquad (32)$$

or also:

$$\begin{bmatrix}\dot{\mathbf{q}}_p\\ \mathbf{t}_p\end{bmatrix} = \begin{bmatrix}-\mathbf{J}_{t_p} & \mathbf{J}_p\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{J}_{t_a} & \mathbf{J}_{t_e}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_e\end{bmatrix} = \begin{bmatrix}\mathbf{J}^s_{11} & \mathbf{J}^s_{12}\\ \mathbf{J}^s_{21} & \mathbf{J}^s_{22}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_e\end{bmatrix} \qquad (33)$$

The case where the robot is overconstrained is not detailed here for reasons of text compactness, but a method similar to the one proposed in [20] can be used.

Finally,

$$\begin{bmatrix}\dot{\mathbf{q}}_t\\ \mathbf{t}_p\end{bmatrix} = \begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_p\\ \dot{\mathbf{q}}_e\\ \mathbf{t}_p\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{J}^s_{11} & \mathbf{J}^s_{12}\\ \mathbf{0} & \mathbf{I}\\ \mathbf{J}^s_{21} & \mathbf{J}^s_{22}\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_a\\ \dot{\mathbf{q}}_e\end{bmatrix} = \mathbf{J}\dot{\mathbf{q}} \qquad (34)$$

C. Computation of the passive joints and platform accelerations

Differentiating (31) with respect to time and using (19) leads to:

$$\mathbf{J}_p\dot{\mathbf{t}}_p + \dot{\mathbf{J}}_p\mathbf{t}_p = \mathbf{J}_t\ddot{\mathbf{q}}_t + \begin{bmatrix}{}^0\mathbf{g}^T_{1m_1}, \cdots, {}^0\mathbf{g}^T_{nm_n}\end{bmatrix}^T = \mathbf{J}_t\ddot{\mathbf{q}}_t + \mathbf{g}_{tot} \qquad (35)$$

Rearranging leads to:

$$\begin{bmatrix}\ddot{\mathbf{q}}_p\\ \dot{\mathbf{t}}_p\end{bmatrix} = \mathbf{J}^s\ddot{\mathbf{q}} + \begin{bmatrix}-\mathbf{J}_{t_p} & \mathbf{J}_p\end{bmatrix}^{-1}\left(\mathbf{g}_{tot} - \dot{\mathbf{J}}_p\mathbf{t}_p\right) = \mathbf{J}^s\ddot{\mathbf{q}} + \begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{bmatrix} \qquad (36)$$

Finally:

$$\begin{bmatrix}\ddot{\mathbf{q}}_t\\ \dot{\mathbf{t}}_p\end{bmatrix} = \mathbf{J}\ddot{\mathbf{q}} + \begin{bmatrix}\mathbf{0} & \mathbf{b}^T_1 & \mathbf{0} & \mathbf{b}^T_2\end{bmatrix}^T = \mathbf{J}\ddot{\mathbf{q}} + \mathbf{b} \qquad (37)$$

D. Elastodynamic model of the actual parallel robot

Introducing (37) into (30) leads to

$$\begin{bmatrix}\tau\\ \mathbf{0}_{n_e}\end{bmatrix} = \mathbf{J}^T\mathbf{M}_t\mathbf{J}\ddot{\mathbf{q}} + \mathbf{J}^T\left(\mathbf{c}_t + \mathbf{M}_t\mathbf{b}\right) = \mathbf{M}\ddot{\mathbf{q}} + \mathbf{c} \qquad (38)$$

which is the full elastodynamic model of the parallel robot.

E. 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 [8]. If the link ij into consideration is flexible, $3N_{ij}$ supplementary elastic variables are introduced ($\mathbf{q}_{e_{ij}}$, $\dot{\mathbf{q}}_{e_{ij}}$ and $\ddot{\mathbf{q}}_{e_{ij}}$) in combination with $6N_{ij}$ shape functions. Then, the previously developed equations are used. 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 contained 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 $\ddot{\mathbf{q}}_e$ in the case of the inverse model, $\ddot{\mathbf{q}}_a$ and $\ddot{\mathbf{q}}_e$ 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.

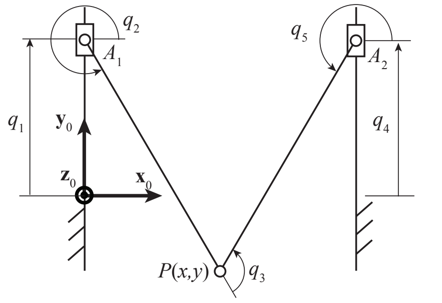


Fig. 3. The $\underline{P}RRR\underline{P}$ mechanism.

V. ILLUSTRATIVE EXAMPLE

To illustrate the previous equations, a flexible planar $\underline{P}RRR\underline{P}$ mechanism is modelled (Fig. 3 - *R* stands for a passive revolute joint and $\underline{P}$ for an active prismatic joint). Its two actuated prismatic pairs are parallel. Its modified Denavit-Hartenberg parameters are given in Table I, where $l = A_1P = A_2P$ and d is the distance between the two prismatic axes. A mass of 1 kg is added on the end-effector (body 3, whose origin is point P) and each actuator mass is equal to 1 kg. The gravity is directed along $\mathbf{z}_0$. The friction effects are not introduced into the model. The beams are modelled as planar three dimension finite elements [14] (one element, i.e. three elastic coordinates by beam). Their cross sections are full squares of edge length equal to 2 mm and the used material is aluminum. Of course, with only two beams, the model accuracy with respect to the reality will be low. However, the aim of this example if to cross validate our model with commercial softwares.

TABLE I

MODIFIED DENAVIT-HARTENBERG PARAMETERS OF THE $\underline{P}RRR\underline{P}$ MECHANISM.

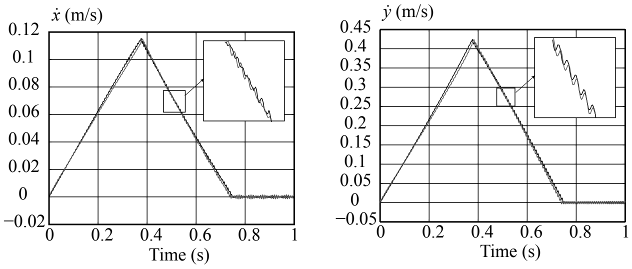
j	$a(j)$	μ_{ji}	σ_{ji}	γ_{ji}	b_{ji}	α_{ji}	d_{ji}	θ_{ji}	r_{ji}
1	0	1	1	0	0	0	0	0	q_1
2	1	0	0	0	0	$\pi/2$	0	q_2	0
3	2	0	0	0	0	0	$l = 0.4m$	q_3	0
4	0	1	1	0	0	0	$a = 0.6m$	0	q_4
5	4	0	0	0	0	$\pi/2$	0	q_5	0

The model is thus calculated with Mathematica applying the proposed methods and then solved using Matlab/Simulink. First, the six natural frequencies of the robot

TABLE II

COMPARISON OF NATURAL FREQUENCIES OF THE *P̲RRRP̲* MECHANISM COMPUTED WITH THE PROPOSED MODEL AND RDM6.

		q_4 (m)				
(Hz)		-0.5	-0.25	0	0.25	0.5
f_1	RDM6	31.416	31.454	31.452	31.454	31.416
	model	31.414	31.454	31.454	31.454	31.414
	% error	0.006	0.000	0.006	0.000	0.006
f_2	RDM6	31.457	31.462	31.452	31.462	31.457
	model	31.455	31.455	31.455	31.455	31.455
	% error	0.006	0.022	0.010	0.022	0.006
f_3	RDM6	40.777	109.650	124.390	109.650	40.777
	model	40.799	109.664	124.389	109.664	40.799
	% error	0.054	0.013	0.001	0.013	0.054
f_4	RDM6	144.150	144.050	140.750	144.050	144.150
	model	144.139	144.041	140.757	144.041	144.139
	% error	0.008	0.006	0.005	0.006	0.008
f_5	RDM6	144.190	144.220	144.210	144.220	144.190
	model	144.181	144.201	144.225	144.201	144.181
	% error	0.006	0.013	0.010	0.013	0.006
f_6	RDM6	183.600	152.960	144.480	152.960	183.600
	model	183.588	152.949	144.496	152.949	183.588
	% error	0.007	0.007	0.011	0.007	0.007



(a) $\dot{x}(t)$

(b) $\dot{y}(t)$

Fig. 4. Velocity of the robot end-effector (black curve: computed model; grey curve: ADAMS model).

are computed and compared for five different robot configurations with the commercial software RDM6 (software able to compute the structure natural frequencies [21]). As the two prismatic axes are parallel, the robot configuration can be defined by only one parameter, here q_4 (q_1 is set to 0). The results are shown in Table II. The error is inferior to 0.1 %: the accuracy of natural frequency computation is very good compared with RDM6.

Then, the full model is compared with a model designed with ADAMS/Flex in which the beams are modelled via a series of discrete flexible links (15 elements by beam). A movement during 0.75s between the intial and final configurations $q_{10} = q_{40} = 0$ and $q_{1f} = 0.1$m, $q_{4f} = 0.2$m is achieved using a bang-bang acceleration profile [8]. The full model is composed of 521 '+' or '-' operators, and of 611 '*' or '/' and runs in about 100 seconds (about 20 min for ADAMS) on a processor Pentium 2.7GHz (8Go of RAM). The resulting velocity of the end-effector is drawn in Fig. 4. It can be observed that the two models give similar results. The observed differences are due to the different approaches used in the beam modelling and to the differences between the ADAMS and Matlab solvers.

VI. CONCLUSIONS

This paper has presented a symbolic and recursive calculation of the dynamic model of flexible parallel robots. 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 method, both link and joint flexibilities can be taken into account.

The proposed algorithm was used to compute the elastodynamic model of a planar parallel robot. The obtained model has been compared with models obtained with commercial softwares and the obtained results have shown its validity.

REFERENCES

[1] W. Khalil and O. Ibrahim, "General solution for the dynamic modeling of parallel robots," *Journal of Intelligent and Robotic Systems*, vol. 49, no. 1, pp. 19–37, 2007.

[2] O. Ibrahim and W. Khalil, "Inverse and direct dynamic models of hybrid robots," *Mechanism and Machine Theory*, vol. 45, pp. 627–640, 2010.

[3] F. Moon, *Applied dynamics*. J. Wiley and Sons, 2007.

[4] R. R. Craig and M. C. C. Bampton, "Coupling of substructures for dynamic analysis," *AIAA Journal*, vol. 6, no. 7, 1968.

[5] 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.

[6] R. Craig, *Structural dynamics*. Wiley, 1981.

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

[8] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*. Hermes Penton London, 2002.

[9] W. Khalil and M. Gautier, "Modeling of mechanical systems with lumped elasticity," in *Proceedings of the IEEE International Conference on Robotics and Automation*, (San Francisco, CA, USA), pp. 3965–3970, 2000.

[10] F. Boyer and W. Khalil, "An efficient calculation of the flexible manipulator inverse dynamics," *International Journal of Robotics Research*, vol. 17, no. 3, pp. 282–293, 1998.

[11] S. Dwivedy and P. Eberhard, "Dynamic analysis of flexible manipulators, a litterature review," *Mechanism and Machine Theory*, vol. 41, no. 7, pp. 749–777, 2006.

[12] J. Kruszewski, W. Gawronski, E. Wittbrodt, F. Najbar, and S. Grabowski, "The rigid finite element method," (Arkady, Warszawa), 1975.

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

[14] A. Shabana, *Dynamics of Multibody Systems*. Cambridge University Press, 2005.

[15] O. Bauchau, *Flexible multibody dynamics*. Springer, 2011.

[16] K. Stachera and W. Schumacher, *Automation and Robotics*, ch. 15: Derivation and Calculation of the dynamics of Elastic parallel manipulators. I. Tech. Educ. & Publishing, 2008.

[17] M. Rognant, E. Courteille, and P. Maurine, "A systematic procedure for the elastodynamic modeling and identification of robot manipulators," *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1085–1093, 2010.

[18] J. de Jalon and E. Bayo, *Kinematic and dynamic simulations of multibody systems*. Springer-Verlag.

[19] F. Boyer, W. Khalil, M. Benosman, and G. L. Vey, *Commande des robots manipulateurs*, ch. Modélisation et commande des robots souples, pp. 187–241. Hermes, 2002.

[20] C. Germain, S. Briot, S. Caro, and P. Wenger, "An efficient method for the natural frequency computation of parallel robots," *Multibody System Dynamics*, in press.

[21] "http://iut.univ-lemans.fr/ydlogi/."