Recursive and Symbolic Calculation of the Stiffness and Mass Matrices of Parallel Robots

Sébastien Briot and Wisama Khalil

Abstract This paper presents a symbolic and recursive calculation of the stiffness and mass matrices of 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 model expression. Some algorithms have been proposed for the rigid case or for parallel robots with lumped springs. In this paper, we extend the previous works to parallel robots with distributed flexibilities. The proposed algorithm, that takes advantage of recursive calculations for the computation of the Jacobian matrices defining the kinematic constraints, is used to compute the natural frequencies of a robot developed at IRCCyN: the NaVARo.

Key words: Parallel robots, Flexibilities, Natural frequencies.

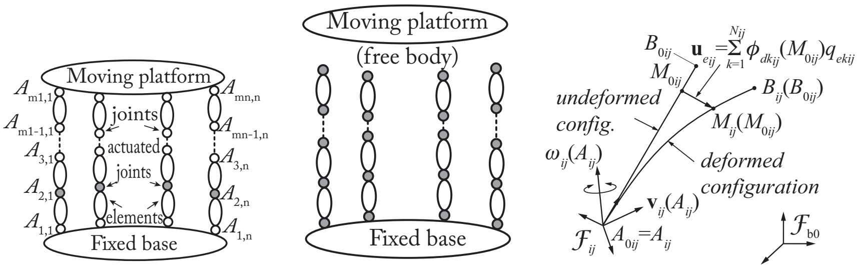
1 Introduction

The large computational time required for calculating the natural frequencies of a robot prevents to use them in many applications, such as real-time control, design optimization process, etc. To decrease the computational cost, this paper focuses on the efficient symbolic computation for the stiffness and mass matrices of flexible parallel robots. This approach could be combined with model reduction methods [6, 9, 10], but this is not the main goal of that paper.

For the computation of the robot natural frequencies, two main approaches are generally proposed (see [11] for a large literature review): (i) lumped modeling [13, 18] and (ii) modeling using distributed flexibilities [2, 3, 7, 16, 17]. The lumped modeling is generally simpler to use by non-experts in finite element meth-

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

Sébastien Briot and Wisama Khalil
IRCCyN, UMR CNRS 6597, France, and École Centrale de Nantes, France
e-mail: {Sebastien.Briot,Wisama.Khalil}@irccyn.ec-nantes.fr



(a) Kinematic chain (b) Virtual tree structure (c) A single flexible link

Fig. 1 A general parallel robot composed of flexible elements.

ods but, to obtain a correct model accuracy, higher number of elements is required, thus increasing the computational time.

Contrary to lumped modeling, using distributed flexibilities allows the improvement of the model accuracy. However, such methods require highly-skilled users. In [2, 16, 17], some general methodologies are proposed. In the case of closed-loop mechanisms, some Jacobian matrices are computed that allow taking into account the kinematic dependencies. However, such general methodologies are not specifically designed for parallel robots and they do not guarantee 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 mass and stiffness matrices (using distributed flexibilities) of parallel robots with a minimal numbers of operators has never been proposed.

The present work aims at filling this gap. In order to minimize the number of operations, the Jacobian matrices defined in the principle of virtual powers (*PVP*) are computed using recursive algorithms. For computing the stiffness and mass matrices of parallel robots, the approach proposed in [5] is adapted. It proposes to (1) convert the parallel robot into a virtual system defined by a tree-structure robot composed of the kinematic chains of the actual robot for which all joints (passive and active) are considered actuated and a free body (the platform which is considered as rigid) (Fig. 1), (2) compute the elastodynamic model of this new virtual system, and (3) finally, close the loops by using the *PVP*.

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

2 Stiffness and Mass Matrices for the Virtual System

Let us consider a parallel robot composed of one rigid fixed base (denoted as element 0), one rigid moving platform and n legs, each leg being a serial kinematic chain composed of $(m_i - 1)$ elements[1] connected by m_i joints of coordinates q_{ik}

[1] Note that each robot link can be composed of one element or several elements.

(revolute, prismatic or fixed joints – $i = 1,...,n$) located at points A_{ik} ($k = 1,...,m_i$ - Fig. 1(a)). The j-th element of the i-th leg is denoted by ij and its displacement can be parameterized by the coordinates ${}^{ij}\mathbf{t}_{ij}$[2] which represents the twist of the body ij at the origin of the local frame $\mathscr{F}_{ij}$ (Fig. 1(c)) and $\dot{\mathbf{q}}_{e_{ij}}$ that are the generalized velocities characterizing the elastic displacement of the body ij

$$ {}^{ij}\mathbf{t}_{e_{ij}}(M_{ij}) = \boldsymbol{\Phi}_{ij}(M_{0ij})\dot{\mathbf{q}}_{e_{ij}} \tag{1}$$

where ${}^{ij}\mathbf{t}_{e_{ij}}(M_{ij})$ is the deformation twist due to the body elasticity that can be parameterized using truncated series of Rayleigh-Ritz shape functions $\boldsymbol{\Phi}_{ij}$ [4].

The vector of generalized coordinates of the tree-structure is given by $\mathbf{q}_t = \left[\mathbf{q}_{t_1}^T \cdots \mathbf{q}_{t_n}^T\right]^T$, where $\mathbf{q}_{t_i}$ regroups all joint variables (denoted as $\mathbf{q}_{p_i}^T = [q_{i1} \ldots q_{im_i}]$) and elastic generalized coordinates $\mathbf{q}_{e_i}^T = \left[\mathbf{q}_{e_{i1}}^T \ldots \mathbf{q}_{e_{i,m_i}}^T\right]$ for the real i-th leg.

The Lagrangian of the tree structure system can be expressed as:

$$ L_t = \frac{1}{2}\sum_{i,j}\left(\begin{bmatrix} {}^{ij}\mathbf{t}_{ij}^T & \dot{\mathbf{q}}_{e_{ij}}^T \end{bmatrix} \mathbf{M}_{ij} \begin{bmatrix} {}^{ij}\mathbf{t}_{ij} \\ \dot{\mathbf{q}}_{e_{ij}} \end{bmatrix} - \mathbf{q}_{e_{ij}}^T \mathbf{K}_{ij} \mathbf{q}_{e_{ij}}\right) \tag{2}$$

where $\mathbf{M}_{ij}$ and $\mathbf{K}_{ij}$ are the mass and stiffness matrices of the link ij whose full expressions in the most compact form are given in [4,15]. To express the Lagrangian as a function of $\mathbf{q}_t$ and $\dot{\mathbf{q}}_t$, let us express the displacement of the element ij frame located at A_{ij} using the following equations obtained by a recursive algorithm [4]

$$ {}^{ij}\mathbf{t}_{ij} = \mathbf{J}_{t_{ij}}\dot{\mathbf{q}}_t \text{ with } \mathbf{J}_{t_{ij}} = {}^{ij}\mathbf{T}_{i(j-1)}\mathbf{J}_{t_{i(j-1)}} + \boldsymbol{\Phi}_{\mathbf{q}_{eij}} + \mathbf{A}_{ij} \text{ and} \tag{3}$$

$$ \boldsymbol{\Phi}_{\mathbf{q}_{eij}} = \begin{bmatrix} \mathbf{0} \cdots {}^{ij}\overline{\mathbf{R}}_{i(j-1)}\boldsymbol{\Phi}_{i(j-1)}(A_{ij}) \cdots \mathbf{0} \end{bmatrix}, \mathbf{A}_{ij} = \begin{bmatrix} \mathbf{0} \cdots {}^{ij}\mathbf{a}_{ij} \cdots \mathbf{0} \end{bmatrix} \tag{4}$$

where $\mathbf{a}_{ij}$ is the unit twist describing the joint ij axis [4] and, in $\boldsymbol{\Phi}_{\mathbf{q}_{eij}}$ ($\mathbf{A}_{ij}$, resp.), the term ${}^{ij}\overline{\mathbf{R}}_{i(j-1)}\boldsymbol{\Phi}_{i(j-1)}(A_{ij})$ (${}^{ij}\mathbf{a}_{ij}$, resp.) is located at the columns corresponding to the variables $\dot{\mathbf{q}}_{e_{i(j-1)}}$ ($\dot{q}_{ij}$, resp.). Moreover, in the previous expressions,

$$ {}^{ij}\overline{\mathbf{R}}_{i(j-1)} = \begin{bmatrix} {}^{ij}\mathbf{R}_{i(j-1)} & \mathbf{0} \\ \mathbf{0} & {}^{ij}\mathbf{R}_{i(j-1)} \end{bmatrix}, {}^{ij}\mathbf{T}_{i(j-1)} = {}^{ij}\overline{\mathbf{R}}_{i(j-1)} \begin{bmatrix} \mathbf{I_3} & -{}^{i(j-1)}\hat{\mathbf{r}}_{i(j-1)}(A_{ij}) \\ \mathbf{0} & \mathbf{I_3} \end{bmatrix} \tag{5}$$

where $\mathbf{I_3}$ is the (3×3) identity matrix, ${}^{ij}\mathbf{R}_{i(j-1)}$ is the rotation matrix between frames $\mathscr{F}_{ij}$ and $\mathscr{F}_{i(j-1)}$, ${}^{i(j-1)}\hat{\mathbf{r}}_{i(j-1)}(A_{ij})$ 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 $A_{ij} \equiv B_{i(j-1)}$ in $\mathscr{F}_{i(j-1)}$ (Fig. 1(c)).

Finally, a global Jacobian matrix $\mathbf{J}_{ij}$ defined such as $\begin{bmatrix} {}^{ij}\mathbf{t}_{ij}^T \dot{\mathbf{q}}_{e_{ij}}^T \end{bmatrix}^T = \mathbf{J}_{ij}\dot{\mathbf{q}}_t$ can be computed as $\mathbf{J}_{ij}^T = \begin{bmatrix} \mathbf{J}_{t_{ij}}^T & \mathbf{O}_{\mathbf{q}_{eij}}^T \end{bmatrix}$ where $\mathbf{O}_{\mathbf{q}_{eij}}$ is defined such that $\dot{\mathbf{q}}_{e_{ij}} = \mathbf{O}_{\mathbf{q}_{eij}}\dot{\mathbf{q}}_t$.

Introducing $\begin{bmatrix} {}^{ij}\mathbf{t}_{ij}^T \dot{\mathbf{q}}_{e_{ij}}^T \end{bmatrix}^T = \mathbf{J}_{ij}\dot{\mathbf{q}}_t$ into (2) leads to

[2] In what follows, the preceeding superscript "ij" denotes that the vector is given in the frame $\mathscr{F}_{ij}$.

$$L_t = \frac{1}{2}\sum_{i,j}\left(\dot{\mathbf{q}}_t^T\mathbf{J}_{ij}^T\mathbf{M}_{ij}\mathbf{J}_{ij}\dot{\mathbf{q}}_t - \mathbf{q}_t^T\mathbf{O}_{\mathbf{q}_{eij}}^T\mathbf{K}_{ij}\mathbf{O}_{\mathbf{q}_{eij}}\mathbf{q}_t\right) = \frac{1}{2}\left(\dot{\mathbf{q}}_t^T\mathbf{M}_t\dot{\mathbf{q}}_t - \mathbf{q}_t^T\mathbf{K}_t\mathbf{q}_t\right) \qquad (6)$$

where $\mathbf{M}_t$ and $\mathbf{K}_t$ are the mass and stiffness matrices of the tree structure.

Adding the contribution of the rigid platform into (6), the Lagrangian of the total system can be written as:

$$L = \frac{1}{2}\left(\begin{bmatrix}\dot{\mathbf{q}}_t\\ \mathbf{t}_p\end{bmatrix}^T\begin{bmatrix}\mathbf{M}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_p\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_t\\ \mathbf{t}_p\end{bmatrix} - \begin{bmatrix}\mathbf{q}_t\\ \mathbf{x}_p\end{bmatrix}^T\begin{bmatrix}\mathbf{K}_t & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{q}_t\\ \mathbf{x}_p\end{bmatrix}\right) = \frac{1}{2}\left(\dot{\mathbf{q}}_g^T\mathbf{M}_g\dot{\mathbf{q}}_g - \mathbf{q}_g^T\mathbf{K}_g\mathbf{q}_g\right) \qquad (7)$$

$\mathbf{M}_p$ is the mass matrix of the rigid platform and $\mathbf{x}_p$ represents the platform displacement ($\mathbf{t}_p$ its twist). $\mathbf{M}_g$ and $\mathbf{K}_g$ are the total mass and stiffness matrices of the virtual system. $\mathbf{q}_g^T = \left[\mathbf{q}_t^T\ \mathbf{x}_p^T\right]$ is its vector of generalized coordinates.

3 Stiffness and Mass Matrices for the Parallel Robot

It is now necessary to determine one possible subset of generalized coordinates for the parallel robot. Using (3) for computing the twist ${}^{i,m_i}\mathbf{t}_{i,m_i}$ of the tip of leg i:

$${}^{i,m_i}\mathbf{t}_{i,m_i} = \mathbf{J}_{t_{i,m_i}}^i\dot{\mathbf{q}}_{t_i} \qquad (8)$$

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 $\dot{\mathbf{q}}_{t_i}^T = \left[\dot{\mathbf{q}}_{p_i}^T, \dot{\mathbf{q}}_{e_i}^T\right]$, i.e. the vector stacking all 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:

$${}^{i,m_i}\mathbf{t}_{i,m_i} = \mathbf{J}_p^i\mathbf{t}_p \text{ , where } \mathbf{J}_p^i = {}^{i,m_i}\overline{\mathbf{R}}_0\begin{bmatrix}\mathbf{I}_3 & -{}^0\hat{\mathbf{p}}_i\\ \mathbf{0} & \mathbf{I}_3\end{bmatrix} \qquad (9)$$

in which ${}^0\hat{\mathbf{p}}_i$ is the cross product matrix of vector ${}^0\mathbf{p}_i$ that characterizes the position of the attachment point A_{i,m_i} wrt the platform center (Fig. 1(a)) and ${}^{i,m_i}\overline{\mathbf{R}}_0$ is the (6×6) rotation matrix between the global frame and the local frame $\mathscr{F}_{i,m_i}$.

Thus, the following set of equations can be obtained:

$$\begin{bmatrix}\mathbf{J}_{t_{1,m_1}}^1 & \cdots & \mathbf{0}\\ \vdots & \ddots & \vdots\\ \mathbf{0} & \cdots & \mathbf{J}_{t_{n,mn}}^n\end{bmatrix}\begin{bmatrix}\dot{\mathbf{q}}_{t_1}\\ \vdots\\ \dot{\mathbf{q}}_{t_n}\end{bmatrix} - \begin{bmatrix}\mathbf{J}_p^1\\ \vdots\\ \mathbf{J}_p^n\end{bmatrix}\mathbf{t}_p = \mathbf{0} \Leftrightarrow \mathbf{J}_t\dot{\mathbf{q}}_t - \mathbf{J}_p\mathbf{t}_p = \left[\mathbf{J}_t\ -\mathbf{J}_p\right]\begin{bmatrix}\dot{\mathbf{q}}_t\\ \mathbf{t}_p\end{bmatrix} = \mathbf{J}_g\dot{\mathbf{q}}_g = \mathbf{0} \qquad (10)$$

where $\mathbf{J}_g$ is a $(r\ n \times n_{q_g})$ matrix, $n_{q_g} > r\ n$ ($r = 6$ 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 $\dot{\mathbf{q}}_g$ is linked to the others. This subset is not unique. As most of parallel robots have identical legs, an idea is to put in $\dot{\mathbf{q}}_d$ the last r components $\dot{\mathbf{q}}_{t_i}^f$ of each vector $\dot{\mathbf{q}}_{t_i}$ which can be decomposed into two parts $\dot{\mathbf{q}}_{t_i}^T = \left[\dot{\mathbf{q}}_{t_i}^{0T}\ \ \dot{\mathbf{q}}_{t_i}^{fT}\right]$:

$$-\begin{bmatrix} \mathbf{J}^{f1}_{t1,m_1} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}^{fn}_{tn,mn} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^{f}_{t1} \\ \vdots \\ \dot{\mathbf{q}}^{f}_{tn} \end{bmatrix} = \begin{bmatrix} \mathbf{J}^{01}_{t1,m_1} & \cdots & \mathbf{0} & -\mathbf{J}^{1}_{p} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{J}^{0n}_{tn,mn} & -\mathbf{J}^{n}_{p} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}^{0}_{t1} \\ \vdots \\ \dot{\mathbf{q}}^{0}_{tn} \\ \mathbf{t}_p \end{bmatrix} \Leftrightarrow \mathbf{J}^{f}_{t}\dot{\mathbf{q}}^{f}_{t} = \mathbf{J}_{tp}\dot{\mathbf{q}} \quad (11)$$

This can be rewritten as

$$\dot{\mathbf{q}}^{f}_{t} = \left(\mathbf{J}^{f}_{t}\right)^{-1}\mathbf{J}_{tp}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_{d_{1,1}} & \cdots & \mathbf{J}_{d_{1,n}} & \mathbf{J}_{d_{1,n+1}} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{J}_{d_{n,1}} & \cdots & \mathbf{J}_{d_{n,n}} & \mathbf{J}_{d_{n,n+1}} \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_d\dot{\mathbf{q}} \quad (12)$$

If the coordinates $\dot{\mathbf{q}}^{f}_{t_i}$ are those of the last elastic element of the leg (which is most often the case), the k-th column of matrix $\mathbf{J}^{fi}_{til}$ corresponds to a unit twist that describes the displacement of the leg extremity due to the k-th coordinate of $\mathbf{q}^{f}_{t_i}$, i.e. [12]

$$\mathbf{J}^{fi}_{til} = \begin{bmatrix} {}^{i,m_i}\mathbf{R}_{il} & -{}^{i,m_i}\mathbf{R}_{il}{}^{il}\hat{\mathbf{p}}_{il} \\ \mathbf{0} & {}^{i,m_i}\mathbf{R}_{il} \end{bmatrix} \quad (13)$$

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

$$\left(\mathbf{J}^{fi}_{til}\right)^{-1} = \begin{bmatrix} {}^{i,m_i}\mathbf{R}^{T}_{il} & \hat{\mathbf{p}}_{il}{}^{i,m_i}\mathbf{R}^{T}_{il} \\ \mathbf{0} & {}^{i,m_i}\mathbf{R}^{T}_{il} \end{bmatrix} \quad (14)$$

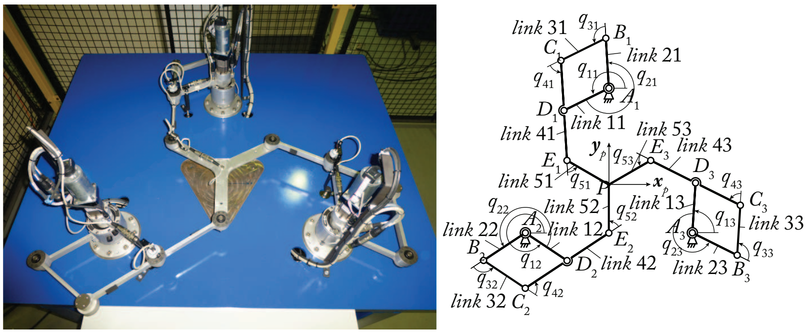
which does not require much calculation. Finally, from (12) and the definition of $\dot{\mathbf{q}}^{T}_{g} = \left[\dot{\mathbf{q}}^{T}_{t_1},\ \ldots,\ \dot{\mathbf{q}}^{T}_{t_n},\ \mathbf{t}^{T}_{p}\right]^{T}$, the matrix $\mathbf{J}$ defined such that $\dot{\mathbf{q}}_g = \mathbf{J}\dot{\mathbf{q}}$ can be computed.

Introducing $\dot{\mathbf{q}}_g = \mathbf{J}\dot{\mathbf{q}}$ into (7) leads to:

$$L = \frac{1}{2}\left(\dot{\mathbf{q}}^{T}\mathbf{J}^{T}\mathbf{M}_g\mathbf{J}\dot{\mathbf{q}} - \mathbf{q}^{T}\mathbf{J}^{T}\mathbf{K}_g\mathbf{J}\mathbf{q}\right) = \frac{1}{2}\left(\dot{\mathbf{q}}^{T}\mathbf{M}\dot{\mathbf{q}} - \mathbf{q}^{T}\mathbf{K}\mathbf{q}\right) \quad (15)$$

from which the natural frequencies f_i $(i = 1, \ldots, n_{q_g} - r\,n)$ of the parallel robot can be computed as $f_i = \sqrt{eig\left(\mathbf{M}^{-1}\mathbf{K}\right)}/(2\pi)$. It should be noticed that obviously, matrices $\mathbf{M}$ and $\mathbf{K}$ depend on the robot configuration.

To automatize the calculation of the mass and stiffness matrices of the robot, 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 model is obtained as a set of intermediate variables. Those that have no effect on the desired output can be eliminated. This algorithm has been successfully implemented with Mathematica.



(a) Prototype of the NaVARo

(b) Shematics of the NaVARo

Fig. 2 The NaVARo

4 Case Study: the NaVARo

The NaVARo is a 3-*dof* planar parallel manipulator developed at IRCCyN (Fig. 2(a)) and composed of 3 identical legs and one moving platform made up of 3 segments E_1P, E_2P and E_3P rigidly linked at point P. The i-th leg contains four links connected with five revolute joints in such a way that $A_iB_iC_iD_i$ is a parallelogram linkage, $i = 1,2,3$. The base frame $\mathscr{F}_b\,(O,\mathbf{x}_0,\mathbf{y}_0,\mathbf{z}_0)$ (not shown in Fig. 2(b)) is defined such as point O is located at the geometric centre of the equilateral triangle $A_1A_2A_3$. Frame $\mathscr{F}_p\,(P,\mathbf{x}_p,\mathbf{y}_p,\mathbf{z}_p)$ is attached to the moving platform. In the home configuration shown in Fig. 2, $\mathscr{F}_b$ and $\mathscr{F}_p\,(P,\mathbf{x}_p,\mathbf{y}_p,\mathbf{z}_p)$ (attached to the moving platform) coincide. (x_p, y_p) are the Cartesian coordinates of point P expressed in frame $\mathscr{F}_b$ and θ_p is the orientation angle of the platform (the angle between $\mathbf{x}_0$ and $\mathbf{x}_p$).

Three double clutches are mounted to the base at points A_i, $i = 1,2,3$, in order to actuate either angle q_{1i} or angle q_{2i}. As a consequence, the NaVARo has 8 actuation modes [1,14]. Therefore, the platform can be moved throughout the workspace without reaching any parallel singularity thanks to a judicious actuation scheme. The kinematics of the i-th leg is described by the modified Denavit-Hartenberg parameters (*MDH*) [12] given in Table 1, in which $\gamma_i = \pi/2$ if $i = 1$, $\gamma_i = -5\pi/6$ if $i = 2$ and $\gamma_i = -\pi/6$ if $i = 3$. Besides, the circumradius of the moving-platform is equal to 0.2027 m, i.e., $l_{5i} = 0.2027$ m. Each link, of rectangular cross-section, is made up of duraluminum alloy. Table 1 gives their cross-section area and inertia.

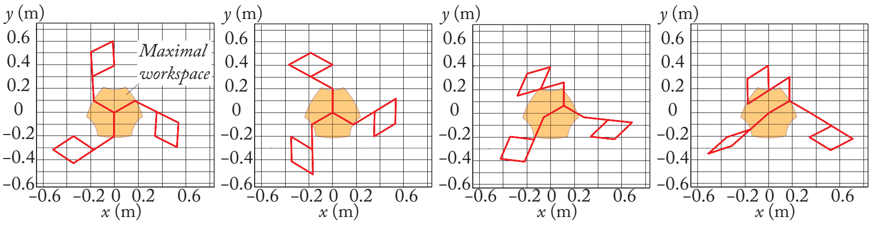
In the experimental setup, the rotation of links $1i$ and $2i$ about point A_i, $i = 1,2,3$, is locked thanks to the double clutch mechanisms. A single 3D beam element is used to model links $1i$, $2i$, $3i$ and $5i$ while two 3D beam elements of equal lengths are used to model links $4i$, the latter being twice longer than the former.

Thus, the NaVARo is modelled as a spatial mechanism and its model contains 144 generalized coordinates, among which only 90 are independent (see Sect. 3). The model has been calculated using the proposed procedure and compiled into C code to obtain the robot natural frequencies. The computation involve the use of 36183 ‘+’ or ‘-’ and 37341 ‘*’ or ‘/’ operators, while 21383 variables are defined.

Table 1 *MDH* parameters of the *i*-th leg and characteristics of the beam cross-sections

ji	$a(ji)$	σ_{ji}	γ_{ji}	b_{ji}	α_{ji}	d_{ji}	θ_{ji}	r_{ji}
$1i$	0	0	γ_i	0	0	$d_1 = 0.4041\text{m}$	$q_{1i} - \gamma_i$	0
$2i$	0	0	γ_i	0	0	$d_1 = 0.4041\text{m}$	$q_{2i} - \gamma_i$	0
$3i$	$2i$	0	0	0	0	$d_3 = 0.2100\text{m}$	q_{3i}	0
$4i$	$3i$	0	0	0	0	$d_4 = 0.2100\text{m}$	q_{4i}	0
$5i$	$4i$	0	0	0	0	$d_5 = 0.4200\text{m}$	q_{5i}	0

link	A_{ij} (m^2)	$I_{y_{ij}}$ (m^4)	$I_{z_{ij}}$ (m^4)	$I_{p_{ij}}$ (m^4)	$I_{0_{ij}}$ (m^4)
$1i$, $2i$, $3i$, $4i$	$2.4 \cdot 10^{-4}$	$1.152 \cdot 10^{-8}$	$2.000 \cdot 10^{-9}$	$1.352 \cdot 10^{-8}$	$5.902 \cdot 10^{-9}$
$5i$	$4 \cdot 10^{-4}$	$3.333 \cdot 10^{-8}$	$5.333 \cdot 10^{-8}$	$8.666 \cdot 10^{-8}$	$1.123 \cdot 10^{-8}$



(a) $\{0,0,0\}$ (b) $\{0,0,-\pi/3\}$ (c) $\{0.12,0.07,-\pi/3\}$ (d) $\{0.18,0.11,-\pi/3\}$

Fig. 3 The four test poses $\{x_p, y_p, \theta_p\}$ (positions in meters, orientation in radian)

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, the obtained frequencies were validated by means of an equivalent model developed using Cast3M software [8]. For the simulations, Cast3M gives the result after around 6 sec. of computation while our model send the results in around 0.01 sec. (for a Pentium 4 2.70GHz, 8Go of RAM). Both models give the same values for the first 90 natural frequencies of the NaVARo. Table 2 gives the first 5 natural frequencies of the NaVARo for the 4 robot postures shown in Fig. 3.

5 Conclusions

The paper has presented a symbolic and recursive calculation of the stiffness and mass matrices of parallel robots. The proposed algorithm, that takes advantage of recursive calculations for the computation of the Jacobian matrices defining the kinematic constraints, is used to compute the natural frequencies of a robot developed at IRCCyN: the NaVARo. Results have shown that the proposed model was able to give the same values as a FEA software for the first 90 natural frequencies of the NaVARo but in a considerably reduces computational time (around 0.01 sec. for our model while FEA results were obtained in several sec.).

Table 2 Comparison of the natural frequencies obtained with Cast3M and Matlab.

(Hz)	f_1		f_2		f_3		f_4		f_5	
	Cast3M	Matlab	Cast3M	Matlab	Cast3M	Matlab	Cast3M	Matlab	Cast3M	Matlab
Pose (a)	44.10	44.10	44.10	44.10	53.98	53.98	60.63	60.63	95.62	95.62
Pose (b)	45.71	45.71	45.71	45.71	54.58	54.58	65.35	65.35	97.92	97.92
Pose (c)	36.98	36.98	49.31	49.31	53.37	53.37	67.28	67.28	91.80	91.80
Pose (d)	40.17	40.17	50.32	50.32	52.99	52.99	67.36	67.36	91.52	91.52

References

1. Arakelian, V., Briot, S., Glazunov, V.: Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. Mechanism and Machine Theory **43**(9), 1129–1140 (2008)
2. Bauchau, O.: Flexible multibody dynamics. Springer (2011)
3. Boyer, F., Khalil, W.: An efficient calculation of the flexible manipulator inverse dynamics. International Journal of Robotics Research **17**(3), 282–293 (1998)
4. Boyer, F., Khalil, W., Benosman, M., LeVey, G.: Robot Manipulators. Modeling, Performance, Analysis and Control, chap. 7: Modeling and Control of Flexible Robots, pp. 337–394. Control Systems, Robotics and Manufacturing Series. ISTE (2007)
5. Briot, S., Khalil, W.: Recursive and symbolic calculation of the elastodynamic model of flexible parallel robots. The International Journal of Robotics Research (2014). In press
6. Briot, S., Pashkevich, A., Chablat, D.: Reduced elastodynamic modelling of parallel robots for the computation of their natural frequencies. In: World Congress in Mechanism and Machine Science, 13 (2011)
7. Cammarata, A., Condorelli, D., Sinatra, R.: An algorithm to study the elastodynamics of parallel kinematic machines with lower kinematic pairs. ASME Transactions Journal of Mechanisms and Robotics **5**(1) (2013)
8. Castem3000: The castem software webpage. www-cast3m.cea.fr. URL www-cast3m.cea.fr. Webpage accessed November 2012
9. Craig, R.: Structural dynamics. Wiley (1981)
10. Craig, R.R., Bampton, M.C.C.: Coupling of substructures for dynamic analysis. AIAA Journal **6**(7) (1968)
11. Dwivedy, S., Eberhard, P.: Dynamic analysis of flexible manipulators, a litterature review. Mechanism and Machine Theory **41**(7), 749–777 (2006)
12. Khalil, W., Dombre, E.: Modeling, Identification and Control of Robots. Hermes Penton London (2002)
13. Khalil, W., Gautier, M.: Modeling of mechanical systems with lumped elasticity. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3965–3970. San Francisco, CA, USA (2000)
14. Rakotomanga, N., Chablat, D., Caro, S.: Kinetostatic performance of a planar parallel mechanism with variable actuation. In: Advances in Robot Kinematics (2008)
15. Shabana, A.: Dynamics of flexible bodies using generalized newton-euler equations. Journal of Dynamic Systems, Measurement, and Control **112**, 496–503 (1990)
16. Shabana, A.: Dynamics of Multibody Systems. Cambridge University Press (2005)
17. Stachera, K., Schumacher, W.: Automation and Robotics, chap. 15: Derivation and Calculation of the Dynamics of Elastic Parallel Manipulators. I. Tech. Educ. & Publishing (2008)
18. Wittbrodt, E., Adamiec-Wójcik, I., Wojciech, S.: Dynamics of Flexible Multibody Systems. Springer-Verlag Berlin Heidelberg (2006)