MULTIBODY DYNAMICS 2005, ECCOMAS Thematic Conference
C.L. Bottasso, P. Masarati, L. Trainelli (eds.)
Milano, Italy, 25–28 June 2007

# EFFICIENCY OF TOPOLOGICAL AND GLOBAL FORMULATIONS FOR SMALL AND LARGE FLEXIBLE MULTIBODY SYSTEMS

**Urbano Lugrís[*], Javier Cuadrado[*], Francisco González[*] and Alberto Luaces[*]**

[*]Departamento de Ingeniería Industrial II
University of La Coruña, Escuela Politécnica Superior, Mendizábal, s/n, 15403 Ferrol, Spain
e-mails: ulugris@udc.es, javicuad@cdf.udc.es, fgonzalez@udc.es,
aluaces@udc.es, web page: http://lim.ii.udc.es/index.en.html

**Keywords:** Flexible Multibody Dynamics, Topological, Semi-Recursive, Real-Time.

**Abstract.** *The simulation of flexible multibody systems is a very demanding task that needs improvements in efficiency in order to achieve real-time performance. One of the improvements can be the use of topological formulations, which have provided good results in the simulation of large rigid multibody systems. In this work, a topological formulation for rigid bodies is extended to the flexible case, and tests are carried out in order to compare its performance with that of a global formulation. Three systems are simulated, a double four-bar mechanism, a vehicle suspension, and a full vehicle. As it happened in the rigid case for the first two examples, the topological formulation shows lower performance than its global counterpart for such small systems, but the difference decreases as more bodies are modeled as flexible. Also like in the rigid case, the topological formulation is faster and more robust in the third example, whose size is approximately one order of magnitude larger, though its advantage over the global approach is not as remarkable as in the rigid case. This is due to the calculation of the mass matrices of the flexible bodies that takes most of the CPU-time. The use of a different method for this calculation may have a very significant impact in the performance, so further tests should be conducted in order to verify this point.*

## 1 INTRODUCTION

During the last years, several efficient methods for the real-time simulation of rigid multibody systems have been developed. The performance enhancement experienced by computers makes the inclusion of new features possible, like flexibility and contact. But all these new features have a negative effect on efficiency, making the achievement of real-time simulation more difficult, so that new efforts must be carried out regarding the dynamic formulations, in order to perform more realistic, real-time simulations.

Most real-time methods for the dynamics of rigid multibody systems take advantage of the mechanism topology and, therefore, are called topological (they use relative coordinates). Although more difficult to implement than global methods (those using Cartesian or fully Cartesian coordinates), they have proven to be more efficient for large systems [1].

The purpose of this work is to extend a topological formulation for rigid bodies to the flexible case, and to compare its behavior with that of a global formulation, as it has been previously done for rigid multibody systems.

## 2 GLOBAL AND TOPOLOGICAL METHODS

Two methods, one global and another topological, are considered in this work. The global method is a floating frame of reference (FFR) formulation [2] based on natural coordinates, as described in [3]. The new topological method combines the approach used to address flexibility by the global one, with a rigid body topological semi-recursive formulation which has showed excellent results for large rigid multibody systems [1].

The global FFR flexible formulation is based on natural coordinates. Each flexible body has a local frame of reference attached to it, which is defined by a point at the origin and three orthogonal unit vectors along the axes. This frame experiences the large amplitude motion, and deformations are added on local coordinates, by using component mode synthesis to reduce the model size.

In the rigid case, the topological method, semi-recursive, opens the closed loops to get the associated open loop mechanism, and defines such system with relative coordinates. In order to obtain the dynamic equations, an intermediate set of global Cartesian coordinates is defined at body level (three translations plus three rotations), and then a velocity transformation is carried out to project the equations into the relative coordinates. This projection is recursively performed by accumulation of forces and inertias, taking advantage of the mechanism topology. Then, closed loop conditions must be imposed through the corresponding constraints, implemented in natural global coordinates.

In both methods, the equations of motion, stated through an index-3 augmented Lagrangian formulation, are combined with the integrator (trapezoidal rule), to produce a nonlinear algebraic system of equations with the dependent positions as unknowns [4], solved by Newton-Raphson iteration. Once convergence is attained into the time-step at position level, velocities and accelerations are projected for them to satisfy the first and second derivatives of the constraints.

## 3 THE PROPOSED FORMULATION

The equations of motion, according to an index-3 augmented Lagrangian formulation in relative dependent coordinates, are stated in the form,

$$\mathbf{M}\ddot{\mathbf{z}} + \mathbf{\Phi}_{\mathbf{z}}^{T}\alpha\mathbf{\Phi} + \mathbf{\Phi}_{\mathbf{z}}^{T}\boldsymbol{\lambda}^{*} = \mathbf{Q} \tag{1}$$

where $\mathbf{z}$ is the relative coordinates vector, $\mathbf{M}$ is the mass matrix, $\mathbf{\Phi}$ is the closed-loop constraints vector, $\mathbf{\Phi_z}$ is its Jacobian matrix, $\mathbf{Q}$ is the vector of elastic, externally applied and velocity-dependent forces, and $\boldsymbol{\lambda}^*$ is the Lagrange multipliers vector, obtained from an iteration process carried out within each time step,

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha \boldsymbol{\Phi}_{i+1} \qquad i = 0,1,2,... \tag{2}$$

which starts with $\boldsymbol{\lambda}_0^*$ equal to the value of $\boldsymbol{\lambda}^*$ obtained in the previous time-step.

### 3.1 Flexible body modeling

The position of an arbitrary point r of a deformed body is defined as follows,

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{A}\overline{\mathbf{r}} = \mathbf{r}_0 + \mathbf{A}\left(\overline{\mathbf{r}}_u + \overline{\mathbf{q}}_f\right) \tag{3}$$

where $\mathbf{r}_0$ stands for the position of the origin of the local frame of reference, $\mathbf{A}$ is a rotation matrix defined by the three orthogonal unit vectors of the reference frame [$\mathbf{u}$|$\mathbf{v}$|$\mathbf{w}$], $\overline{\mathbf{r}}_u$ is the undeformed position of the point in local coordinates, and $\overline{\mathbf{q}}_f$ is its local elastic displacement.

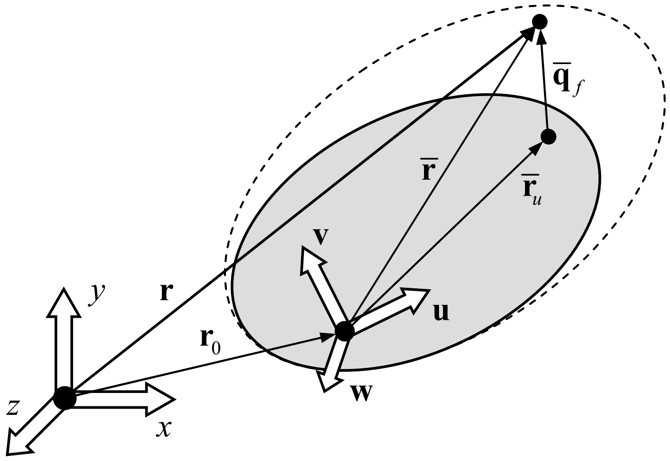

Figure 1: General flexible body.

The elastic deformation is obtained by means of a finite element model of the body, which generally contains a large number of degrees of freedom (DOFs), so that a model reduction must be carried out to reduce computation times. In the proposed method, a Craig-Bampton reduction [5] is used, since it allows for an easy coupling between bodies, and is particularly well suited for a topological implementation, as it will be seen later. This reduction method approximates the elastic displacement field by a linear combination of static and dynamic modes, which can be pre-computed using any finite element code.

Following the formalism of the natural coordinates, the body is connected to the rest of the mechanism by means of boundary points and unit vectors. Boundary points and vectors are associated to finite element displacement and rotation DOFs respectively, so that each static mode is obtained as the displacement field resulting from applying a unit variation to one of these boundary DOFs while keeping the remaining fixed. Dynamic modes are normal eigenmodes calculated with a fixed interface configuration. All static and dynamic modes are calculated with the body clamped at the local frame of reference.

At this point, the topological implementation shows some particular characteristics. In the global formulation, the frame of reference terms $\mathbf{r}_0$ and $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, are problem variables, while in the topological one they are calculated when the kinematic problem is recursively solved for the open loop system, at each time-step, to obtain the joint positions. This avoids the need for additional constraints to ensure that the three vectors are orthonormal. Moreover, compatibility constraints are no longer needed to link the static modal amplitudes to the boundary points and unit vectors, since they are directly obtained while solving the kinematics.

Using this reduction, the local elastic displacement that appears in Eq. (3) can be expressed as a linear combination of deformation modes,

$$\bar{\mathbf{q}}_f = \sum_i^{ns} \mathbf{\Phi}_i \eta_i + \sum_j^{nd} \mathbf{\Psi}_j \xi_j \tag{4}$$

where $\mathbf{\Phi}_i$ and $\mathbf{\Psi}_j$ are the static and dynamic modes, and $\eta_i$ and $\xi_j$ are their respective modal amplitudes, which are added as new coordinates of the multibody system. This expression can be written in matrix form,

$$\bar{\mathbf{q}}_f = \begin{bmatrix} \mathbf{\Phi}_1 & \cdots & \mathbf{\Phi}_{ns} & \mathbf{\Psi}_1 & \cdots & \mathbf{\Psi}_{nd} \end{bmatrix} \begin{Bmatrix} \eta_1 \\ \cdots \\ \eta_{ns} \\ \xi_1 \\ \cdots \\ \xi_{nd} \end{Bmatrix} = \mathbf{X}\mathbf{y} \tag{5}$$

being $\mathbf{X}$ a matrix formed by the modes as columns, and $\mathbf{y}$ a vector containing all the modal amplitudes of the body.

### 3.2 Positions and velocities

The Topological methods cut the closed loops to establish recursive relationships. These loops are then closed by the corresponding kinematic constraints. To calculate the body dynamic terms, an intermediate global Cartesian coordinate vector $\mathbf{Z}$ is defined at velocity level for each body,

$$\mathbf{Z} = \left\{ \dot{\mathbf{s}}^T \quad \boldsymbol{\omega}^T \right\}^T \tag{6}$$

In this vector, $\dot{\mathbf{s}}$ is the velocity of the point of the body which instantly coincides with the origin of the global frame of reference, considering the point as rigidly attached to the body local frame [6], and $\boldsymbol{\omega}$ is the angular velocity vector of the local frame of reference. These coordinates are related to the rigid body motion, and the global velocity vector for a flexible body, after the addition of the modal amplitudes, is,

$$\dot{\mathbf{q}} = \left\{ \dot{\mathbf{s}}^T \quad \boldsymbol{\omega}^T \quad \dot{\mathbf{y}}^T \right\}^T = \left\{ \mathbf{Z}^T \quad \dot{\mathbf{y}}^T \right\}^T \tag{7}$$

The rigid body velocity vector $\mathbf{Z}_i$ of an element $i$ can be obtained for an open loop mechanism by means of a recursive relationship [7],

$$\mathbf{Z}_i = \mathbf{Z}_{i-1} + \mathbf{b}_i \dot{\mathbf{z}}_i + \boldsymbol{\varphi}_{i-1,i} \dot{\boldsymbol{\eta}}_{i-1,i} - \boldsymbol{\varphi}_{i,i} \dot{\boldsymbol{\eta}}_{i,i} \tag{8}$$

To illustrate this expression, Fig. (2) shows a sample planar revolute joint between two deformed bodies $i$-1 and i. $\mathbf{Z}_{i-1}$ is the absolute velocity of the preceding body frame in the kinematic chain. The $\mathbf{z}_i$ vector contains all the relative coordinates defined at joint $i$, so each $\mathbf{b}_i$ matrix column is the relative velocity that arises when giving a unit velocity to the corresponding relative coordinate. In the example, $z_i$ is the relative angle at the joint. Vectors $\boldsymbol{\eta}_{i-1,i}$ and $\boldsymbol{\eta}_{i,i}$ are the amplitudes of the static modes defined at joint $i$ for bodies $i$-1 and $i$ respectively (i.e. the local elastic displacement vectors of the boundary point) so that the $\boldsymbol{\varphi}$ terms

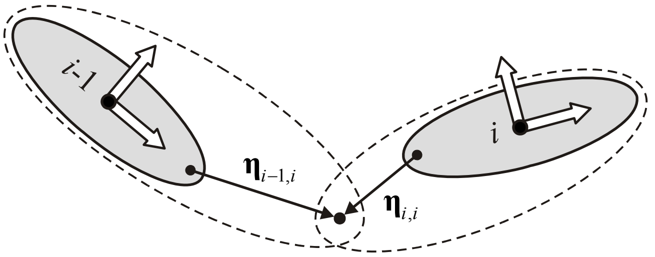

Figure 2: Recursive kinematics in a planar revolute joint.

have an analogous meaning to the **b** ones due to how the static modes are defined. The **b** and **φ** terms depend on the joint and mode type (translational or rotational) respectively, and are functions of the positions. In the particular case of a body whose frame of reference is placed at its entry point, the last term in Eq. (8) doesn't appear, since the body is clamped at its local frame origin.

Accelerations can be calculated by time differentiation of Eq. (8),

$$\dot{\mathbf{Z}}_i = \dot{\mathbf{Z}}_{i-1} + \mathbf{b}_i \ddot{\mathbf{z}}_i + \boldsymbol{\varphi}_{i-1,i} \ddot{\boldsymbol{\eta}}_{i-1,i} - \boldsymbol{\varphi}_{i,i} \ddot{\boldsymbol{\eta}}_{i,i} + \mathbf{d}_i + \dot{\boldsymbol{\varphi}}_{i-1,i} \dot{\boldsymbol{\eta}}_{i-1,i} - \dot{\boldsymbol{\varphi}}_{i,i} \dot{\boldsymbol{\eta}}_{i,i} \tag{9}$$

where $\mathbf{d}_i$ stands for $\dot{\mathbf{b}}_i \dot{\mathbf{z}}_i$.

Equations (8) and (9) yield the velocities and accelerations of the local frames of reference. Differentiation of Eq. (3) particularized to a finite element node *n* yields its velocity $\mathbf{v}_n^*$ as a function of its absolute position $\mathbf{r}_n^*$ and the body coordinates **q**,

$$\mathbf{v}_n^* = \dot{\mathbf{s}} + \boldsymbol{\omega} \times \mathbf{r}_n^* + \mathbf{A}\left(\mathbf{X}_n^* \dot{\mathbf{y}}\right) \tag{10}$$

where $\mathbf{X}_n^*$ is the submatrix formed by the three rows of **X** corresponding to the elastic displacements of node *n*. This expression can be written in matrix form to include the velocities of all the *nn* nodes,

$$\mathbf{v}^* = \begin{Bmatrix} \mathbf{v}_1^* \\ \mathbf{v}_2^* \\ \cdots \\ \mathbf{v}_{nn}^* \end{Bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\tilde{\mathbf{r}}_1^* & \mathbf{A}\mathbf{X}_1^* \\ \mathbf{I}_3 & -\tilde{\mathbf{r}}_2^* & \mathbf{A}\mathbf{X}_2^* \\ \cdots & \cdots & \cdots \\ \mathbf{I}_3 & -\tilde{\mathbf{r}}_{nn}^* & \mathbf{A}\mathbf{X}_{nn}^* \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{s}} \\ \boldsymbol{\omega} \\ \dot{\mathbf{y}} \end{Bmatrix} = \mathbf{B}\dot{\mathbf{q}} \tag{11}$$

where the tilde denotes the skew-symmetric matrix associated to the corresponding vector. The velocities of all the nodes, required for the calculation of the inertia terms, are therefore expressed as a function of the body coordinates **q**.

### 3.3 Dynamic terms in Cartesian coordinates

The kinetic energy of a body can be expressed as,

$$T = \frac{1}{2} \int_V \dot{\mathbf{r}}^T \dot{\mathbf{r}} dm \tag{12}$$

where *V* is the volume of the deformed body. For the calculation of this integral, the co-rotational approximation proposed by Géradin and Cardona [8] is used. This approximation, which leads to a simplified mass matrix at the cost of introducing a kinematic inconsistency, assumes that the finite element interpolation functions **N**, intended for the interpolation of elastic displacements, can be used as well for the interpolation of velocities. In that case, the kinetic energy can be approximated in terms of the finite element mass matrix $\mathbf{M}_{FEM}$ and the nodal velocities $\mathbf{v}^*$,

$$T = \frac{1}{2}\int_V \mathbf{v}^{*T}\mathbf{N}^T\mathbf{N}\mathbf{v}^* dm = \frac{1}{2}\mathbf{v}^{*T}\mathbf{M}_{FEM}\mathbf{v}^* \tag{13}$$

These nodal velocities can be calculated by using the **B** matrix defined in Eq. (11), so that the mass matrix $\bar{\mathbf{M}}$ of an elastic body in body coordinates is obtained as,

$$T = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{B}^T\mathbf{M}_{FEM}\mathbf{B}\dot{\mathbf{q}} \Rightarrow \bar{\mathbf{M}} = \mathbf{B}^T\mathbf{M}_{FEM}\mathbf{B} \tag{14}$$

This is a very simple expression for the mass matrix, taking into account that the finite element mass matrix is directly obtained from a standard finite element code and is constant. Application of the Lagrange equations to this expression of the kinetic energy leads to the velocity dependent inertia forces,

$$\bar{\mathbf{Q}}_v = -\mathbf{B}^T\mathbf{M}_{FEM}\dot{\mathbf{B}}\dot{\mathbf{q}} \tag{15}$$

The elastic potential of a deformed body is obtained from the finite element stiffness matrix $\mathbf{K}_{FEM}$ and the nodal elastic displacements,

$$U = \frac{1}{2}\bar{\mathbf{q}}_f^{*T}\mathbf{K}_{FEM}\bar{\mathbf{q}}_f^* \tag{16}$$

The elastic displacements of the nodes from Eq. (5) can be introduced into this equation, so that a stiffness matrix **K** is obtained in terms of the modal amplitudes,

$$U = \frac{1}{2}\mathbf{y}^T\mathbf{X}^T\mathbf{K}_{FEM}\mathbf{X}\mathbf{y} \Rightarrow \mathbf{K} = \mathbf{X}^T\mathbf{K}_{FEM}\mathbf{X} \tag{17}$$

and this constant matrix can be used for the calculation of the elastic forces,

$$\bar{\mathbf{Q}}_{el} = -\frac{\partial U}{\partial \mathbf{q}} = -\mathbf{K}\mathbf{y} \tag{18}$$

Applied forces are introduced in body coordinates by means of the virtual power principle. For example, for a point force **F** applied at node $\mathbf{r}_i$,

$$\bar{\mathbf{Q}}_{ext}^T\delta\dot{\mathbf{q}} = \mathbf{F}^T\delta\dot{\mathbf{r}}_i \Rightarrow \bar{\mathbf{Q}}_{ext} = \left(\frac{\partial \mathbf{r}_i}{\partial \mathbf{q}}\right)^T\mathbf{F} = \mathbf{B}_i^T\mathbf{F} \tag{19}$$

where $\mathbf{B}_i$ is the three row submatrix of **B** corresponding to the three degrees of freedom of node *i*.

### 3.4 Equations of motion

In order to make the following steps clearer, the coordinates of the whole system will be grouped into two vectors, one for each coordinate set,

$$\begin{aligned}\dot{\mathbf{q}} &= \left\{\mathbf{Z}_1^T \quad \cdots \quad \mathbf{Z}_{nb}^T \quad \dot{\boldsymbol{\eta}}_1^T \quad \cdots \quad \dot{\boldsymbol{\eta}}_{nb}^T \quad \dot{\boldsymbol{\xi}}_1^T \quad \cdots \quad \dot{\boldsymbol{\xi}}_{nb}^T\right\}^T \\ \dot{\mathbf{z}} &= \left\{\dot{z}_1 \quad \cdots \quad \dot{z}_{nc} \quad \dot{\boldsymbol{\eta}}_1^T \quad \cdots \quad \dot{\boldsymbol{\eta}}_{nb}^T \quad \dot{\boldsymbol{\xi}}_1^T \quad \cdots \quad \dot{\boldsymbol{\xi}}_{nb}^T\right\}^T\end{aligned} \tag{20}$$

where *nb* stands for the number of bodies in the system, and *nc* is the number of relative coordinates of the open loop version of the mechanism. The mass matrix and force vector in body coordinates can be assembled for the whole system, having separate blocks for the rigid body part, the static modal amplitudes, and the dynamic modal amplitudes,

$$\bar{\mathbf{M}} = \begin{bmatrix} \bar{\mathbf{M}}_{RB} & \bar{\mathbf{M}}_{RB,\eta} & \bar{\mathbf{M}}_{RB,\xi} \\ \bar{\mathbf{M}}_{\eta,RB} & \bar{\mathbf{M}}_{\eta} & \bar{\mathbf{M}}_{\eta,\xi} \\ \bar{\mathbf{M}}_{\xi,RB} & \bar{\mathbf{M}}_{\xi,\eta} & \bar{\mathbf{M}}_{\xi} \end{bmatrix} \quad ; \quad \bar{\mathbf{Q}} = \begin{Bmatrix} \bar{\mathbf{Q}}_{RB} \\ \bar{\mathbf{Q}}_{\eta} \\ \bar{\mathbf{Q}}_{\xi} \end{Bmatrix} \tag{21}$$

The application of the virtual power principle yields,

$$\dot{\mathbf{q}}^{*T}\left(\bar{\mathbf{M}}\ddot{\mathbf{q}} - \bar{\mathbf{Q}}\right) = 0 \tag{22}$$

equation that must be transformed into the set of relative coordinates **z**. They are related to the body coordinates **q** by a transformation matrix **R** such that,

$$\dot{\mathbf{q}} = \mathbf{R}\dot{\mathbf{z}} \tag{23}$$

Substitution of **q** and its derivatives in Eq. (22) results in the following expression for the equations of motion, taking into account that the **z** coordinates are independent for an open loop system,

$$\mathbf{R}^T\bar{\mathbf{M}}\mathbf{R}\ddot{\mathbf{z}} = \mathbf{R}^T\left(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right) \tag{24}$$

This means that the leading matrix and the right hand side of the equations of motion in relative coordinates are,

$$\mathbf{M} = \mathbf{R}^T\bar{\mathbf{M}}\mathbf{R} \quad ; \quad \mathbf{Q} = \mathbf{R}^T\left(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right) \tag{25}$$

These operations can be performed very efficiently by taking advantage of the open loop topology. The **R** matrix is the result of assembling in matrix form the recursive relationships defined in Eq. (8) for the open loop system, which makes its structure rather particular. Matrix **R** can be divided into blocks if the **q** and **z** coordinates are arranged as described in Eq. (20),

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{RB} & \mathbf{R}_{\eta} & 0 \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & \mathbf{I} \end{bmatrix} \tag{26}$$

where $\mathbf{R}_{RB}$ and $\mathbf{R}_{\eta}$ are two submatrices which relate the Cartesian coordinates to the relative coordinates and to the static modal amplitudes, respectively. The first submatrix, the rigid body part of **R**, would be the **R** matrix of an equivalent rigid mechanism in the current deformed configuration.

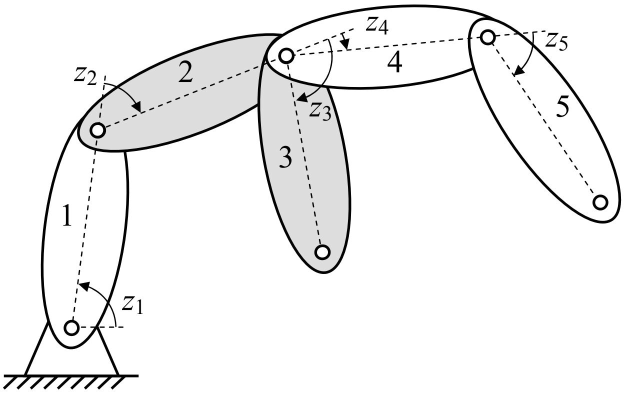

Figure 3: Mechanism topology example.

The example mechanism described in Fig. (3), in which bodies 2 and 3 are flexible, is used to show how the **R** matrix terms look like,

$$\mathbf{R}_{RB}=\begin{bmatrix}\mathbf{I}_6 & 0 & 0 & 0 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & 0 & 0 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{I}_6 & 0 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & 0 & \mathbf{I}_6 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & 0 & \mathbf{I}_6 & \mathbf{I}_6\end{bmatrix}\begin{bmatrix}\mathbf{b}_1 & 0 & 0 & 0 & 0\\ 0 & \mathbf{b}_2 & 0 & 0 & 0\\ 0 & 0 & \mathbf{b}_3 & 0 & 0\\ 0 & 0 & 0 & \mathbf{b}_4 & 0\\ 0 & 0 & 0 & 0 & \mathbf{b}_5\end{bmatrix}=\mathbf{T}_{RB}\mathbf{R}_{RB}^{d}$$

$$\mathbf{R}_{\eta}=\begin{bmatrix}0 & 0 & 0 & 0\\ \mathbf{I}_6 & 0 & 0 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & \mathbf{I}_6 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & 0 & 0\\ \mathbf{I}_6 & \mathbf{I}_6 & 0 & 0\end{bmatrix}\begin{bmatrix}-\boldsymbol{\varphi}_{2,2} & 0 & 0 & 0\\ 0 & \boldsymbol{\varphi}_{2,34} & 0 & 0\\ 0 & 0 & -\boldsymbol{\varphi}_{3,3} & 0\\ 0 & 0 & 0 & 0\end{bmatrix}=\mathbf{T}_{\eta}\mathbf{R}_{\eta}^{d} \tag{27}$$

Each of these two submatrices can be considered as the product of a connectivity matrix **T**, which depends exclusively on the mechanism topology, and a block diagonal matrix $\mathbf{R}^d$, containing the kinematic **b** or **φ** terms. Each block column of the connectivity matrices is associated to either a joint (*RB*) or a boundary generating static modes ($\eta$), and contains identity matrices in the rows corresponding to those bodies being affected by a variation of the column's relative coordinates or modal amplitudes. Expanding in blocks the mass matrix projection of Eq. (25) [7],

$$\mathbf{R}^T\bar{\mathbf{M}}\mathbf{R}=\begin{bmatrix}\mathbf{R}_{RB}^T\bar{\mathbf{M}}_{RB}\mathbf{R}_{RB} & \mathbf{R}_{RB}^T\bar{\mathbf{M}}_{RB}\mathbf{R}_{\eta} & 0\\ \mathbf{R}_{\eta}^T\bar{\mathbf{M}}_{RB}\mathbf{R}_{RB} & \mathbf{R}_{\eta}^T\bar{\mathbf{M}}_{RB}\mathbf{R}_{\eta} & 0\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\ 0 & \bar{\mathbf{M}}_{\eta} & \bar{\mathbf{M}}_{\eta,\xi}\\ 0 & \bar{\mathbf{M}}_{\xi,\eta} & \bar{\mathbf{M}}_{\xi}\end{bmatrix}$$
$$+\begin{bmatrix}0 & \mathbf{R}_{RB}^T\bar{\mathbf{M}}_{RB,\eta} & \mathbf{R}_{RB}^T\bar{\mathbf{M}}_{RB,\xi}\\ \bar{\mathbf{M}}_{\eta,RB}\mathbf{R}_{RB} & \mathbf{R}_{\eta}^T\bar{\mathbf{M}}_{RB,\eta}+\bar{\mathbf{M}}_{\eta,RB}\mathbf{R}_{\eta} & \mathbf{R}_{\eta}^T\bar{\mathbf{M}}_{RB,\xi}\\ \bar{\mathbf{M}}_{\xi,RB}\mathbf{R}_{RB} & \bar{\mathbf{M}}_{\xi,RB}\mathbf{R}_{\eta} & 0\end{bmatrix} \tag{28}$$

The same may be done for the force vector,

$$\mathbf{R}^T\left(\bar{\mathbf{Q}}-\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)=\begin{bmatrix}\mathbf{R}_{RB}^T\left\{\bar{\mathbf{Q}}_{RB}-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_{RB}\right\}+\mathbf{R}_{\eta}^T\left\{\bar{\mathbf{Q}}_{\eta}-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_{\eta}\right\}\\ \bar{\mathbf{Q}}_{\eta}\\ \bar{\mathbf{Q}}_{\xi}\end{bmatrix} \tag{29}$$

All of these terms can be recursively calculated by accumulating the individual mass matrices from the leaves to the root. For example, the rigid body part of the mass matrix

$$\mathbf{R}_{RB}^T\bar{\mathbf{M}}_{RB}\mathbf{R}_{RB}=\mathbf{R}_{RB}^{dT}\left(\mathbf{T}_{RB}^T\bar{\mathbf{M}}_{RB}\mathbf{T}_{RB}\right)\mathbf{R}_{RB}^{d} \tag{30}$$

For the mechanism shown in Fig. (3),

$$\mathbf{T}_{RB}^{T}\bar{\mathbf{M}}_{RB}\mathbf{T}_{RB}=\begin{bmatrix}\mathbf{M}_1 & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5\\ & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5\\ & & \mathbf{M}_3 & 0 & 0\\ & sym & & \mathbf{M}_4 & \mathbf{M}_5\\ & & & & \mathbf{M}_5\end{bmatrix}\quad ;\quad \begin{aligned}\mathbf{M}_5&=\bar{\mathbf{M}}_5\\ \mathbf{M}_4&=\bar{\mathbf{M}}_4+\mathbf{M}_5\\ \mathbf{M}_3&=\bar{\mathbf{M}}_3\\ \mathbf{M}_2&=\bar{\mathbf{M}}_2+\mathbf{M}_3+\mathbf{M}_4\\ \mathbf{M}_1&=\bar{\mathbf{M}}_1+\mathbf{M}_2\end{aligned}\tag{31}$$

where all the *RB* subindices have been removed for clarity. All the remaining terms of the mass matrix can be calculated following a similar procedure, taking advantage on the connectivity matrices. The same procedure is explained to show how the first term of the rigid body part of **Q** is obtained,

$$\mathbf{R}_{RB}^{T}\left\{\bar{\mathbf{Q}}_{RB}-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_{RB}\right\}=\mathbf{R}_{RB}^{dT}\mathbf{T}_{RB}^{T}\left\{\bar{\mathbf{Q}}_{RB}-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_{RB}\right\}\tag{32}$$

For the Fig. (3) example,

$$\mathbf{T}_{RB}^{T}\left\{\bar{\mathbf{Q}}_{RB}-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_{RB}\right\}=\begin{Bmatrix}\mathbf{Q}_1\\ \mathbf{Q}_2\\ \mathbf{Q}_3\\ \mathbf{Q}_4\\ \mathbf{Q}_5\end{Bmatrix}\quad ;\quad \begin{aligned}\mathbf{Q}_5&=\bar{\mathbf{Q}}_5-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_5\\ \mathbf{Q}_4&=\bar{\mathbf{Q}}_4-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_4+\mathbf{Q}_5\\ \mathbf{Q}_3&=\bar{\mathbf{Q}}_3-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_3\\ \mathbf{Q}_2&=\bar{\mathbf{Q}}_2-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_2+\mathbf{Q}_3+\mathbf{Q}_4\\ \mathbf{Q}_1&=\bar{\mathbf{Q}}_1-\left(\bar{\mathbf{M}}\dot{\mathbf{R}}\dot{\mathbf{z}}\right)_1+\mathbf{Q}_2\end{aligned}\tag{33}$$

where the *RB* subindices have been removed again for making the equations smaller.

Once the dynamic terms have been obtained in relative coordinates for the open loop system, the closed loop kinematic constraints are imposed in natural coordinates. The Jacobian matrix of the constraints appearing in Eq. (1) is evaluated by differentiating the constraints with respect to the relative coordinates, which can be done by means of the chain differentiation rule,

$$\mathbf{\Phi}_{\mathbf{z}}=\mathbf{\Phi}_{\mathbf{q}}\mathbf{q}_{\mathbf{z}}\tag{34}$$

where in this case **q** stands for the natural coordinates at the corresponding cut joint. The term $\mathbf{q}_{\mathbf{z}}$ is easily calculated since each column contains the velocities of natural coordinates when a unit velocity is given to its corresponding relative coordinate *z* and zero to the rest.

### 3.5 Time integration

The numerical integrator adopted is the implicit single-step trapezoidal rule, whose difference equations, for a time-step of Δ*t* are,

$$\begin{aligned}\dot{\mathbf{z}}_{n+1}&=\frac{2}{\Delta t}\mathbf{z}_{n+1}-\left(\frac{2}{\Delta t}\mathbf{z}_n+\dot{\mathbf{z}}_n\right)\\ \ddot{\mathbf{z}}_{n+1}&=\frac{4}{\Delta t^2}\mathbf{z}_{n+1}-\left(\frac{4}{\Delta t^2}\mathbf{z}_n+\frac{4}{\Delta t}\dot{\mathbf{z}}_n+\ddot{\mathbf{z}}_n\right)\end{aligned}\tag{35}$$

If dynamic equilibrium is imposed at step *n*+1 by combining the integrator equations (35) with the equations of motion (1), a nonlinear system of algebraic equations must be solved for the relative coordinates in *n*+1,

$$\mathbf{f}\left(\mathbf{z}_{n+1}\right)=0 \tag{36}$$

The system can be solved using the Newton-Raphson iteration with the following approximated tangent matrix and residual vector,

$$\begin{aligned}\mathbf{f}_{\mathbf{z}} &\cong \mathbf{M}+\frac{\Delta t}{2}\mathbf{C}+\frac{\Delta t^2}{4}\left(\mathbf{\Phi}_{\mathbf{z}}^T\alpha\mathbf{\Phi}_{\mathbf{z}}+\mathbf{K}\right)\\ \mathbf{f} &= \frac{\Delta t^2}{4}\left(\mathbf{M}\ddot{\mathbf{q}}+\mathbf{\Phi}_{\mathbf{z}}^T\alpha\mathbf{\Phi}_{\mathbf{z}}+\mathbf{\Phi}_{\mathbf{z}}^T\boldsymbol{\lambda}^*-\mathbf{Q}\right)\end{aligned} \tag{37}$$

being **K** and **C** the generalized stiffness and damping matrices,

$$\mathbf{K}=-\mathbf{Q}_{\mathbf{z}} \quad ; \quad \mathbf{C}=-\mathbf{Q}_{\dot{\mathbf{z}}} \tag{38}$$

which can be calculated using the chain differentiation rule if the forces are expressed in natural coordinates.

The solution of Eq. (36) yields a position vector that fulfills the dynamic equilibrium equations and the kinematic restrictions at position level $\mathbf{\Phi}=0$. However, the velocities and accelerations do not satisfy the derivatives of the constraints, since they have not been imposed. Therefore, the resulting velocities and accelerations need to be projected in order to enforce their fulfillment of the constraint derivatives. Naming $\dot{\mathbf{z}}^*$ and $\ddot{\mathbf{z}}^*$ the velocities and accelerations obtained once the Newton-Raphson iteration has converged, the new projected velocities and accelerations are obtained solving the following linear systems,

$$\begin{aligned}\mathbf{f}_{\mathbf{z}}\dot{\mathbf{z}} &= \left[\mathbf{M}+\frac{\Delta t}{2}\mathbf{C}+\frac{\Delta t^2}{4}\mathbf{K}\right]\dot{\mathbf{z}}^*-\frac{\Delta t^2}{4}\mathbf{\Phi}_{\mathbf{z}}^T\alpha\mathbf{\Phi}_t\\ \mathbf{f}_{\mathbf{z}}\ddot{\mathbf{z}} &= \left[\mathbf{M}+\frac{\Delta t}{2}\mathbf{C}+\frac{\Delta t^2}{4}\mathbf{K}\right]\ddot{\mathbf{z}}^*-\frac{\Delta t^2}{4}\mathbf{\Phi}_{\mathbf{z}}^T\alpha\left(\dot{\mathbf{\Phi}}_{\mathbf{z}}\dot{\mathbf{z}}+\dot{\mathbf{\Phi}}_t\right)\end{aligned} \tag{39}$$

## 4 TEST EXAMPLES

Three examples, already used for a global vs topological comparison in rigid multibody systems [1], have been implemented in the flexible case through both the global and the topological formulation. The first one is a planar double four-bar mechanism, formed by five identical bars, the second is the front left suspension of the Iltis, and the third is a full Iltis vehicle. Performance measurements have been carried out with different numbers of flexible elements, in order to evaluate the influence of such parameter in each formulation. The first two examples were implemented in MATLAB, so the CPU-times should not be considered as a reference for the efficiency, but only for comparison between formulations. The Iltis vehicle is programmed in FORTRAN, obtaining faster simulations despite of being a much larger system.

### 4.1 Double four-bar mechanism

The system consists of five identical steel bars, each of them having unit length and mass, connected by revolute joints. Each bar can be considered as rigid or flexible, modeled in the flexible case by 10 beam elements, with one axial static, one bending static, and two bending dynamic modes.

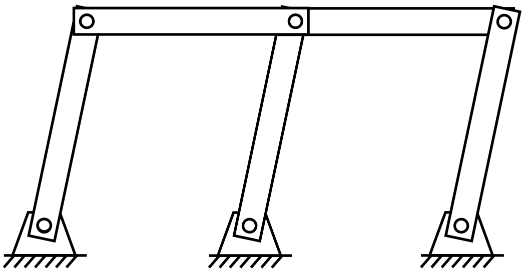

Figure 4: Double four-bar mechanism.

The number of coordinates varies as more flexible bars are considered in the system, being this variation different in the global and topological formulations, because the latter doesn't include as coordinates the unit vectors of the local frames. The number of coordinates for each formulation and number of flexible bodies is given in Table 1.

| # flexible bars | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Global | 6 | 13 | 20 | 27 | 34 | 41 |
| Topological | 5 | 8 | 11 | 14 | 17 | 20 |

Table 1: Number of system coordinates in the first example.

This number tends to be double in the global case when more flexible bodies are considered, because each flexible body adds four modal amplitudes plus four unit vector components, while, in the topological case, each body adds only the four modal amplitudes.

The system is subject to gravity, and receives an initial velocity of 1 rad/s in clockwise direction. Motion is integrated during 5 s, the time to approximately complete 2.7 revolutions. The time-step used in all simulations is 10 ms, and the CPU-times required for the integration are those provided in Table 2.

| # flexible bars | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| Global | 0.91 | 3.30 | 6.24 | 9.61 | 11.51 | 15.22 |
| Topological | 4.85 | 9.11 | 12.62 | 15.74 | 17.74 | 20.92 |

Table 2: CPU-times (s) in the first example.

As it may be seen in the table, the CPU-times reduce their difference when more bodies are considered flexible. In the rigid case, the global formulation is five times faster, while, in the fully flexible model, the topological formulation needs only 37% more time for integration, probably due to the proportionally lower number of coordinates mainly.

### 4.2 Iltis suspension

The front left suspension of the Bombardier Iltis vehicle [9] consists, as shown in Fig. (5), on a lower A-arm triangle (A) connected to both the car body and the wheel hub. A damper connects the A-arm to the car body, and the upper side of the hub is connected to the chassis by means of a leaf spring, which is modeled as an articulated bar (B) with a spring element added between the bar tip and the chassis. In this system, three elements can be considered as flexible: the A-arm, the bar that models the leaf spring, and the steering rod (S).

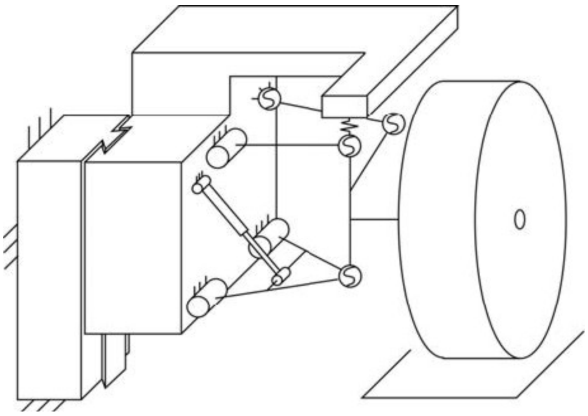

Figure 5: Iltis suspension.

All of them have been modeled as steel elements, with 10 finite elements per bar (two bending static modes and four bending dynamic modes) and 21 finite elements in the case of the A-arm (one vertical static mode in the connection to the hub, another in the connection to the damper, and the first two dynamic modes). The number of coordinates of all the possible combinations of formulation and flexible bodies results,

| Flexible elements | None | A | B | S | A+B | A+S | B+S | All |
|---|---|---|---|---|---|---|---|---|
| Global | 25 | 38 | 40 | 40 | 53 | 53 | 55 | 68 |
| Topological | 8 | 12 | 14 | 14 | 18 | 18 | 20 | 24 |

Table 3: Number of system coordinates in the second example.

Being this a three-dimensional system, the consideration of a body as flexible in the global case can add up to 9 coordinates in addition to the modal amplitudes, since a local reference frame needs three unit vectors, though in practice some of the vectors can be shared between elements thus reducing the total number of variables. In this case the number of coordinates in the global model is around three times larger, and, unlike the previous example, this relation remains almost constant with the number of flexible bodies.

The suspension reaches equilibrium and then runs down a 0.2 m step at $t$=2 s. The integration, with a time-step of 10 ms, is carried out for 5 seconds until the suspension reaches equilibrium again. The time history of the vertical coordinate of the chassis, as well as that of the wheel center, with all possible flexible elements, are plotted in Fig. (6), showing a very good agreement between the two formulations (dotted line for the topological method).

The CPU-times required to carry out the simulation are displayed in Table 4. For the cases of one or two flexible bodies, the mean values of the three different combinations are shown.

| # flexible elements | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| Global | 1.95 | 9.05 | 14.56 | 19.43 |
| Topological | 8.19 | 20.54 | 29.55 | 38.57 |

Table 4: CPU-times (s) in the second example.

In the three-dimensional case, the performance difference is reduced when more flexible bodies are considered, as it happened in the planar case. The CPU-time ratio changes from a value of 4 in the rigid case, to a value of 2 when the three mentioned bodies are modeled as flexible.

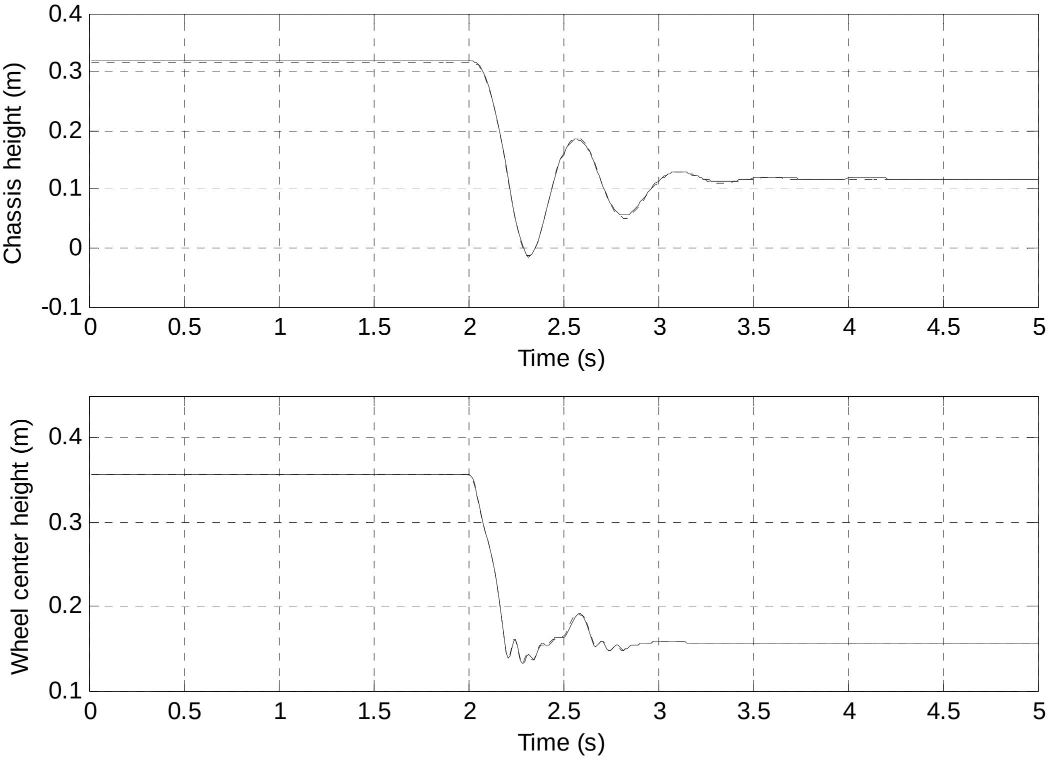

Figure 6: Iltis suspension simulation results.

### 4.3 Iltis vehicle

The Iltis vehicle [9] has four suspensions, all of them identical to the one addressed in the previous example. In this case, modal damping has been added to the flexible elements, using a modal damping matrix equal to 1% of their modal stiffness matrix.

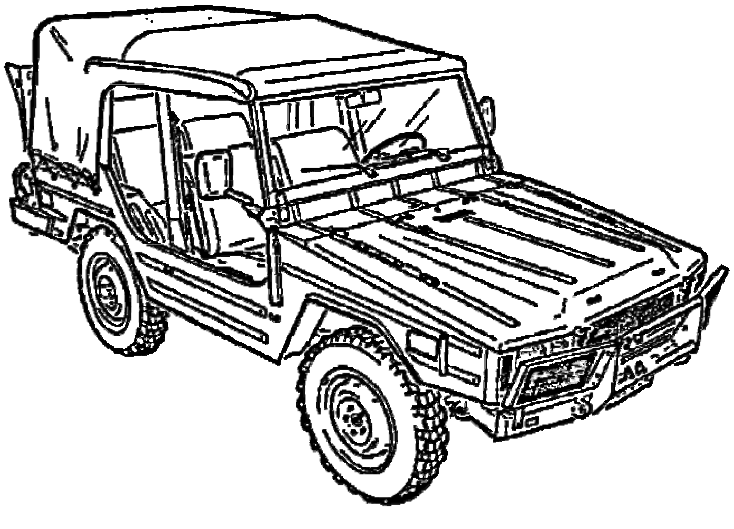

Figure 7: Iltis vehicle.

As it was made in the single suspension problem, all possible combinations of flexible bodies have been tested, using both formulations. The combinations of flexible bodies have been done keeping the modeling of the four suspensions identical, i.e. if one element (A, B or S) is considered as flexible, it is done in all four suspensions. The number of coordinates obtained for each combination is shown in Table 5.

| Flexible elements | None | A | B | S | A+B | A+S | B+S | All |
|---|---|---|---|---|---|---|---|---|
| Global | 168 | 196 | 216 | 228 | 244 | 256 | 276 | 304 |
| Topological | 34 | 50 | 58 | 58 | 74 | 74 | 82 | 98 |

Table 5: Number of system coordinates in the third example.

It is shown that the coordinate numbers tends to be three times lower in the topological case, as it happened in the previous example. But the topological formulation, due to the modal amplitudes, no longer has the very low number of coordinates common for such formulations, reaching a total of 98 coordinates in case of the highest number of flexible bodies.

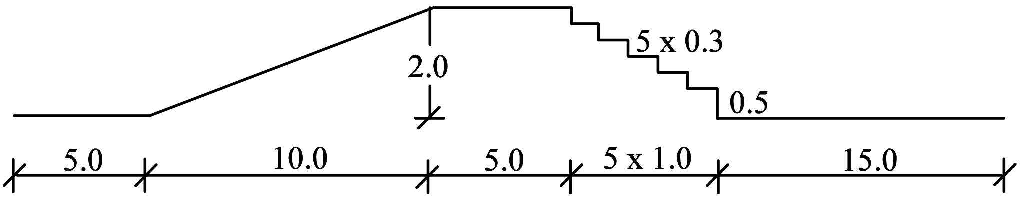

Figure 8: Road profile for the Iltis vehicle simulation.

The vehicle runs during 8 s over the road profile shown in Fig. (8), going from left to right with an initial velocity of 5 m/s. The integration is carried out with a time-step of 10 ms. It must be noted that this is quite a violent maneuver, since the car bounces several times when running down the steps. The time histories of the height of both the chassis origin and the center of the front left wheel, with all flexible bodies, are displayed in the following two figures, showing a very good agreement between the different methods.

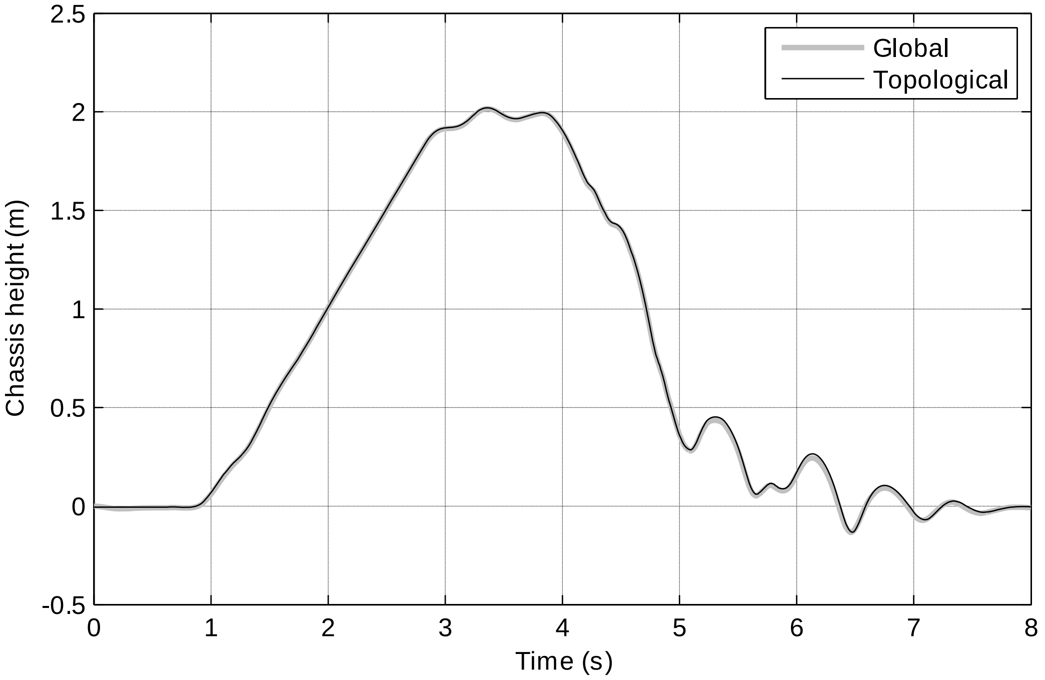

Figure 9: Time-history of the origin of the chassis.

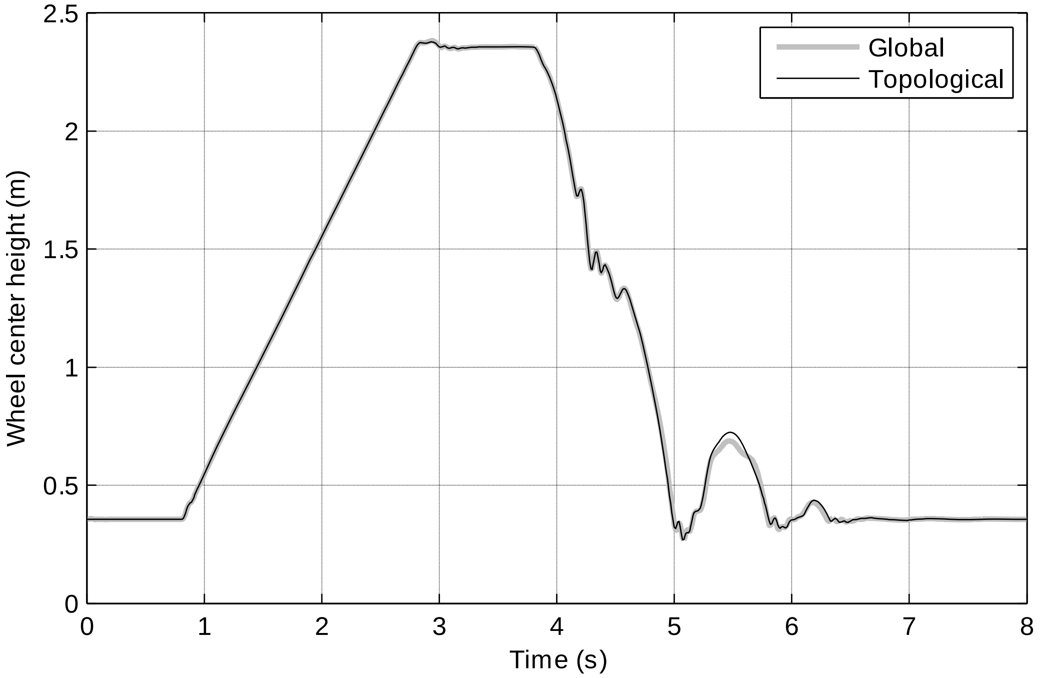

Figure 10: Time-history of the height of the front left wheel.

The integration times obtained can be seen in Table 6 and Fig. (11), including all combinations of flexible and rigid bodies.

| Flexible elements | None | A | B | S | A+B | A+S | B+S | All |
|---|---|---|---|---|---|---|---|---|
| Global | 1.05 | 2.44 | 2.38 | 2.50 | 3.62 | 3.99 | 3.82 | 5.25 |
| Topological | 0.14 | 0.63 | 0.45 | 0.45 | 1.06 | 1.13 | 0.83 | 1.44 |

Table 6: CPU-times (s) in the third example.

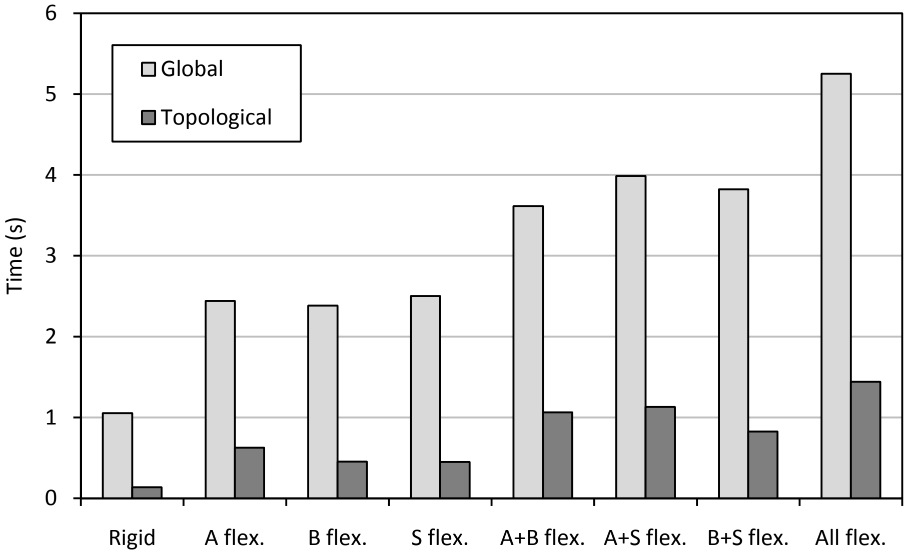

Figure 11: CPU-times (s) in the third example.

Fig. (12) shows the CPU-times vs the number of flexible bodies, in order to show more clearly its influence. The times for four and eight flexible bodies (one and two per suspension) are the mean values of all the corresponding different cases, as in the second example.

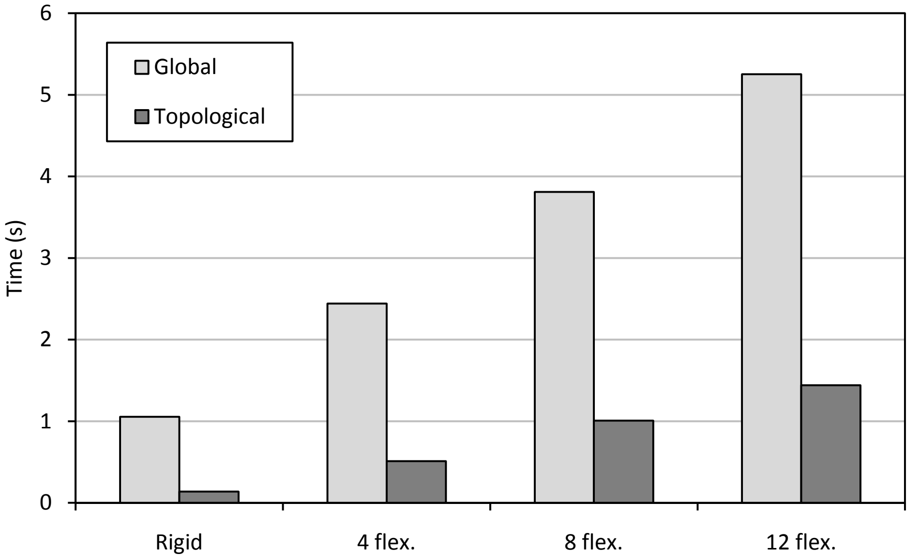

Figure 12: CPU-times (s) vs number of flexible bodies in the third example.

As it happened in the rigid case, the topological method is faster than the global one for large systems, but not to the same extent. As it can be seen in Table 6, the topological method is roughly 8 times faster in the rigid case, but only 4 times in the fully flexible case. This happens because the computation of the flexible mass matrices takes most of the computing time. It should be noted that the topological method needs a very accurate initial position to run properly, while the global method has better tolerance to non-equilibrium initial conditions.

In Table 7 the CPU-times for the fastest simulations possible, obtained by varying the time-step, are shown, along with the corresponding real-time ratios.

| | Global rigid | Global flexible | Topological rigid | Topological flexible |
|---|---|---|---|---|
| Δt (s) | 0.012 | 0.012 | 0.036 | 0.036 |
| CPU-time (s) | 0.896 | 4.607 | 0.078 | 0.953 |
| Real-time ratio | 8.93 | 1.74 | 102.56 | 8.40 |

Table 7: CPU-times (s) in the third example, for highest time-step.

The global method can run with a maximum time-step of 12 ms in both the rigid and the flexible case, obtaining a CPU-time reduction of about 12~15%. The topological formulation allows for an increase of up to 36 ms, with a significant improvement in the CPU-time: 43% in the rigid model and 34% in the flexible one, reaching a real-time ratio of more than 8 times in a system with 12 flexible bodies.

## 5 CONCLUSIONS

A new topological formulation for flexible multibody dynamics has been presented, and its performance compared with that of a global formulation. As expected, the new formulation obtains lower performance than the global one for small systems, and higher performance for large systems, in agreement with the results obtained for the rigid case. However, this time the advantage for large systems is more moderate.

The maximum time-step reached by the topological method in the last example shows that it is not only faster for large systems, but also more robust. This is also demonstrated by the fact that, in order to make the global method run properly the Iltis simulation, it has been necessary to add structural damping to the flexible bodies, although the topological method does not need it and works as well with undamped elements.

The introduction of finite element models through the co-rotational approximation is very easy but has a high impact on performance. Profiling shows that the **B** matrix calculation and mass matrix projection can take most of the total integration time. The impact of these operations obviously grows with the number of flexible bodies, reaching, for the Iltis vehicle with all flexible elements, 82% of the total time in the topological formulation and 72% in the global one. The implementation of a preprocessing stage for obtaining constant mass matrix terms, instead of keeping the size of the underlying finite element model –as it happens with matrix **B**– would affect the results, modifying the difference between formulations. These operations (**B** calculation and mass projection) are performed faster in the topological method, due to the fact that in the global case a flexible body has 12 rigid body coordinates, for only 6 in the topological one. Each of these coordinates adds a column to the **B** matrix, which affects very significantly its size in systems with few modes per flexible body.

The first two examples were implemented in MATLAB, where matrix multiplications are calculated by compiled internal functions and, hence, are very fast, while m-code is very inefficient. This motivated that calculation of the mass matrix and force vector of Eq. (25) by means of the accumulation method described in section 3.4, was slower than by direct multiplication. Consequently, direct multiplication was used in these examples, thus loosing the theoretical advantage provided by the recursive accumulation of masses and forces. Probably, the difference between formulations would be different if these examples were implemented in a compiled language.

The third example has been implemented in FORTRAN, taking advantage of all the possible optimizations applicable to the topological formulation. In the global case, a sparse solver was used, while in the topological formulation a symmetrical dense one was employed, even with matrices as big as 98x98 which could benefit from a sparse solver.

## REFERENCES

[1] J. Cuadrado, D. Dopico, M. González and M.A. Naya, A combined penalty and recursive real-time formulation for multibody dynamics, *Journal of Mechanical Design*, **126**, 602-608, 2004.

[2] A.A. Shabana, *Dynamics of multibody systems*, 2nd edition, Cambridge University Press, Cambridge, 1998.

[3] J. Cuadrado, R. Gutiérrez, M.A. Naya and P. Morer, A comparison in terms of accuracy and efficiency between a MBS dynamic formulation with stress analysis and a non-linear FEA code, *International Journal for Numerical Methods in Engineering*, **51**, 1033–1052, 2001.

[4] J. Cuadrado, J. Cardenal and E. Bayo, Modeling and solution methods for efficient real-time simulation of multibody dynamics, *Multibody System Dynamics*, **1**, 259-280, 1997.

[5] R.R. Craig and M.C.C. Bampton, Coupling of substructures for dynamic analyses, *AIAA Journal*, **6**, 1313-1319, 1968.

[6] D.S. Bae and E.J. Haug, A recursive formulation for constrained mechanical system dynamics. Part I - Open Loop Systems, *Mechanics of Structures and Machines*, **5**, 359-382, 1987.

[7] F.J. Funes, J. García de Jalón, F.A. de Ribera and E. Álvarez, Solution of the dynamics of flexible systems by means of topologic formulations, Proceedings of Métodos Computacionais em Engenharia, Lisbon, 2004. (in Spanish)

[8] M. Géradin and A. Cardona, *Flexible multibody dynamics – A finite element approach*, John Wiley and Sons, New York, 2001.

[9] Iltis Data Package, *IAVSD Workshop*, Herbertov, 1990.