Proceedings of DETC'03
ASME 2003 Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
Chicago, Illinois, USA, September 2-6, 2003

DETC2003/VIB-48315

A HYBRID GLOBAL-TOPOLOGICAL REAL-TIME FORMULATION FOR MULTIBODY SYSTEMS

Javier Cuadrado
Universidad de La Coruña
Escuela Politécnica Superior
Mendizábal s/n, Campus de Esteiro
Ferrol, La Coruña, 15403, Spain
Tel: +34-981337400, Fax: +34-981337410
E-mail: javicuad@cdf.udc.es

Daniel Dopico
Universidad de La Coruña
Escuela Politécnica Superior
Mendizábal s/n, Campus de Esteiro
Ferrol, La Coruña, 15403, Spain
Tel: +34-981337400, Fax: +34-981337410
E-mail: ddopico@mail2.udc.es

ABSTRACT

The continuously improved performance of personal computers enables the real-time motion simulation of complex multibody systems, such as the whole model of an automobile, on a conventional $1,200 PC, provided the adequate formulation is applied. There exist two big families of dynamic formulations, depending on the type of coordinates they use to model the system: global and topological. The former leads to a simple and systematic programming while the latter is very efficient. In this work, a hybrid formulation is presented, obtained by combination of one of the most efficient global formulations and one of the most systematic topological formulations. In this way, it is developed a new formulation which shows, at the same time, easiness of implementation and a high level of efficiency. In order to verify the advantages that the new formulation has over its predecessors, the analysis of four examples is carried out using the three formulations and the corresponding results are compared: a planar mechanism which goes through a singular position, a car suspension with stiff behaviour, and a 6-dof robot with changing configurations, and the full model of a car vehicle. Furthermore, the last example is also analyzed by using a commercial tool, so as to provide the readers with a well-known reference for comparison.

Keywords: multibody systems, real-time dynamics, penalty formulations, semi-recursive formulations, global methods, topological methods

1. INTRODUCTION

Some years ago, the dynamic simulation of complex multibody systems in real-time was an objective difficult to achieve. Not only the fastest formulations available had to be applied, but also powerful and expensive hardware platforms (over $30,000) were needed. Nowadays, thanks to the enormous improvement experimented by the PC performance, either in calculation as well as in graphics, such complex multibody systems as, for example, the full model of a car, can be simulated in real-time on low cost PCs ($1,200). However, the field of multibody dynamics incessantly evolves and, if some years ago the objective was to simulate the motion of a mechanical system consisting of several rigid bodies, today the goal has been put farther, and flexible bodies as well as contact and impact effects should be considered, to mention just some examples. Hence, new super-efficient dynamic algorithms are required, capable of reducing at the minimum the calculation time needed, so that those simulations of complex systems modeled in a realistic way can be achievable.

Methods developed so far for the dynamic analysis of multibody systems can be grouped into two big families: global and topological.

Global methods [1,2] are characterized by the use of a set of coordinates that perfectly defines the position of each body. Due to this fact, the proper dynamic terms (applied and inertia forces) can be independently calculated for each body and, later on, be assembled to form the corresponding terms of the whole mechanism. On the other hand, the kinematic terms (constraint equations which relate the variables) are established in a systematic way for each body and kinematic pair. Consequently, this family of methods leads to simple and general algorithms of easy implementation but not very efficient.

Topological methods [1,2] make use of relative coordinates in order to model the mechanism, so that the position of each body is defined with respect to the previous one in the kinematic chain. This fact invites to take profit of the chain topology to produce algorithms in which the kinematic as well as the dynamic terms are calculated by means of efficient recursive procedures. However, these kind of formulations are usually rather involved and difficult to generalize.

This work is aimed at obtaining a hybrid formulation as combination of one global and another topological, so that the advantages of both types of formulations are kept, while their inherent drawbacks avoided.

2. STARTING FORMULATIONS

Unlike the usual cases described above, two formulations, one global and another topological, have been developed recently that, possessing the advantages of their respective families, avoid the most part of the corresponding drawbacks.

The global method [3] uses natural (global and dependent) coordinates to model the multibody system. It consists of an index-3 augmented Lagrangian formulation, which is combined with the numerical integrator known as the trapezoidal rule, to produce a non-linear algebraic system of equations with the dependent positions as unknowns. Such system is solved through the Newton-Raphson iteration. Once convergence is attained into the time-step, velocities and accelerations are cleaned by means of mass-damping-stiffness-orthogonal projections. The result is a robust and efficient algorithm.

The topological method, called semi-recursive [4], defines a double set of coordinates in the modeling: six coordinates (three translations plus three rotations) for each body, and the relative coordinates of the whole mechanism. The dynamic equations are expressed in the coordinates of the bodies and, then, a velocity projection is carried out which leads to a set of motion equations in the relative coordinates. In order to calculate the leading matrix and the right-hand-side of that set of equations, a recursive technique which accumulates forces and inertias is used. However, as it happens with any topological method, closed-loops should be opened and, later on, the corresponding constraints imposed. Is in this step where the semi-recursive method finds problems, as it chooses to perform a second velocity projection (in order to arrive at a set of motion equations in independent coordinates), which suffers from the usual drawbacks of this technique: range of validity of the independent set of coordinates selected and lack of robustness in singular positions. To avoid differences coming from the integration scheme, the same procedure as for the global formulation is used: the integrator (trapezoidal rule) equations are combined with the motion equations, thus obtaining a non-linear algebraic system of equations where the independent positions are the unknowns, which is solved by means of the Newton-Raphson iteration. The result is an easy and general algorithm.

3. THE PROPOSED FORMULATION

In the proposed approach, the dynamic equations are stated according to the index-3 augmented Lagrangian formulation [3] 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 z are the relative coordinates, $\boldsymbol{M}$ is the mass matrix of the mechanism expressed in terms of the relative coordinates, $\boldsymbol{\Phi}$ is the constraints vector due to the closure conditions of the loops, $\boldsymbol{\Phi}_z$ is the Jacobian matrix of the constraints, α is the penalty factor, $\boldsymbol{Q}$ is the vector of applied and velocity-dependent forces, and $\boldsymbol{\lambda}^*$ is the vector of Lagrange multipliers obtained from the following iteration process (given by sub-index i, while sub-index n stands for the time-step):

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

where the value of $\boldsymbol{\lambda}_0^*$ is taken equal to the $\boldsymbol{\lambda}^*$ worked out in the previous time-step.

In order to determine the dynamic terms $\boldsymbol{M}$ and $\boldsymbol{Q}$, a second set of coordinates is defined. It can be expressed at velocity level for each body of the system in the form,

$$\boldsymbol{Z} = \begin{Bmatrix} \dot{\boldsymbol{s}} \\ \boldsymbol{\omega} \end{Bmatrix} \tag{3}$$

being $\dot{\boldsymbol{s}}$ the velocity of the point of the body which in that particular time is coincident with the fixed frame origin, and $\boldsymbol{\omega}$ the angular velocity of the body.

When expressed in terms of such coordinates, the dynamic terms for a single body are [4],

$$\overline{\boldsymbol{M}} = \begin{bmatrix} m\boldsymbol{I}_3 & -m\tilde{\boldsymbol{g}} \\ m\tilde{\boldsymbol{g}} & \boldsymbol{J} - m\tilde{\boldsymbol{g}}\tilde{\boldsymbol{g}} \end{bmatrix} \tag{4}$$

$$\overline{\boldsymbol{Q}} = \begin{Bmatrix} \boldsymbol{f} - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times m\boldsymbol{g}) \\ \boldsymbol{n} - \boldsymbol{\omega}\times\boldsymbol{J\omega} + \boldsymbol{g}\times(\boldsymbol{f} - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times m\boldsymbol{g})) \end{Bmatrix} \tag{5}$$

wherein m is the body mass, $\boldsymbol{I}_3$ is the 3x3 identity matrix, $\boldsymbol{g}$ is the global position of the mass center of the body, $\tilde{\boldsymbol{g}}$ is the dual anti-symmetric matrix of $\boldsymbol{g}$, $\boldsymbol{J}$ is the inertia tensor of the body with respect to a reference frame parallel to the global one at the mass center of the body, $\boldsymbol{f}$ is the vector of forces applied to the body, and $\boldsymbol{n}$ is the vector of applied moments with respect to the mass center of the body.

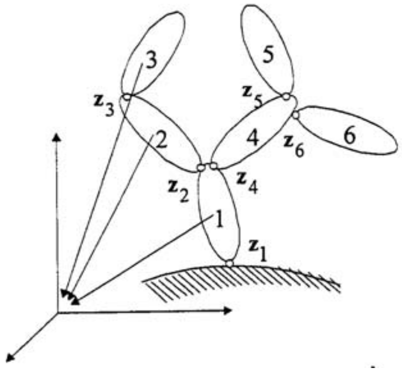

Figure 1. Example of mechanism topology.

A matrix $\boldsymbol{R}$ can be defined so that the following relationship stands,

$$\boldsymbol{Z} = \boldsymbol{R}\dot{\boldsymbol{z}} \tag{6}$$

where now $\boldsymbol{Z}$ includes the body coordinates of all the bodies of the mechanism. Due to the body coordinates adopted, the form of matrix $\boldsymbol{R}$ is rather special, as shown in what follows for the example illustrated in Fig. 1.

$$\boldsymbol{R} = \boldsymbol{TH} \tag{7}$$

with

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_6 & \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_6 & \boldsymbol{I}_6 & \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_6 & \boldsymbol{I}_6 & \boldsymbol{0} \\ \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{I}_6 \end{bmatrix} \tag{8}$$

and,

$$H=\begin{bmatrix} R_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & R_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & R_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & R_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_6 \end{bmatrix} \tag{9}$$

where I_6 is the 6x6 identity matrix and R_i are 6x1 vectors.

If the virtual power principle is applied, the dynamic equations of the system can be written as,

$$Z^{*t}\left(\overline{M}\dot{Z}-\overline{Q}\right)=0 \tag{10}$$

and substituting in Eq. (10) the result of Eq. (6) together with its derivative yields

$$R^t\overline{M}R\ddot{z}=R^t\left(\overline{Q}-\overline{M}\dot{R}\dot{z}\right) \tag{11}$$

which means that the mass matrix of the mechanism expressed in terms of the relative coordinates becomes,

$$M=R^t\overline{M}R \tag{12}$$

and, analogously, the corresponding vector of applied and velocity-dependent forces is,

$$Q=R^t\left(\overline{Q}-\overline{M}\dot{R}\dot{z}\right) \tag{13}$$

If now the special form of matrix R described in Eq. (7-9) is considered, M and Q can be rewritten as,

$$M=H^t\left(T^t\overline{M}T\right)H \tag{14}$$

$$Q=H^t\left(T^t\left(\overline{Q}-\overline{M}T\dot{H}\dot{z}\right)\right) \tag{15}$$

Due to the particular structure of matrices T and H, the mass matrix of Eq. (14) and the force vector of Eq. (15) can be calculated through a very efficient recursive procedure. Hence, for the example of Fig. 1,

$$T^tMT=\begin{bmatrix} M_1 & M_2 & M_3 & M_4 & M_5 & M_6 \\ & M_2 & M_3 & 0 & 0 & 0 \\ & & M_3 & 0 & 0 & 0 \\ & & & M_4 & M_5 & M_6 \\ & sym & & & M_5 & 0 \\ & & & & & M_6 \end{bmatrix} \tag{16}$$

where the sub-matrices are obtained as,

$$\begin{aligned} M_6 &= \overline{M}_6 \\ M_5 &= \overline{M}_5 \\ M_3 &= \overline{M}_3 \\ M_2 &= \overline{M}_2+M_3 \\ M_4 &= \overline{M}_4+M_5+M_6 \\ M_1 &= \overline{M}_1+M_2+M_4 \end{aligned} \tag{17}$$

and,

$$T^t\left(\overline{Q}-\overline{M}T\dot{H}\dot{z}\right)=\begin{Bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \\ Q_6 \end{Bmatrix} \tag{18}$$

where the sub-vectors are obtained as,

$$\begin{aligned} Q_6 &= \overline{Q}_6-\overline{M}_6\left(T\dot{H}\dot{z}\right)_6 \\ Q_5 &= \overline{Q}_5-\overline{M}_5\left(T\dot{H}\dot{z}\right)_5 \\ Q_4 &= \overline{Q}_4-\overline{M}_4\left(T\dot{H}\dot{z}\right)_4+Q_5+Q_6 \\ Q_3 &= \overline{Q}_3-\overline{M}_3\left(T\dot{H}\dot{z}\right)_3 \\ Q_2 &= \overline{Q}_2-\overline{M}_2\left(T\dot{H}\dot{z}\right)_2+Q_3 \\ Q_1 &= \overline{Q}_1-\overline{M}_1\left(T\dot{H}\dot{z}\right)_1+Q_2+Q_4 \end{aligned} \tag{19}$$

So far, the calculation of the M and Q terms of Eq. (1) has been addressed. The remaining term in that equation is the Jacobian matrix of the constraints Φ_z, which implies the differentiation of the constraints vector with respect to the relative coordinates z. This can be easily done by applying the chain differentation rule as,

$$\Phi_z=\Phi_q q_z \tag{20}$$

In the proposed method, natural coordinates [2] are used at the cut-points to impose the closure conditions of the loops. This means that the constraints are expressed in terms of such coordinates, named q in Eq. (20). Therefore, the term Φ_q is the traditional Jacobian matrix of the constraints, when natural coordinates are used [2], while the term q_z simply represents the velocities of the natural coordinates q, when unit velocities are successively given to the relative coordinates z.

Once the calculation of all the terms appearing in Eq. (1) has been explained, the main thread of the proposed formulation can be taken again.

As integration scheme, the implicit single-step trapezoidal rule has been adopted. The corresponding difference equations in velocities and accelerations are:

$$\dot{z}_{n+1}=\frac{2}{\Delta t}z_{n+1}+\hat{\dot{z}}_n \tag{21}$$

$$\ddot{z}_{n+1}=\frac{4}{\Delta t^2}z_{n+1}+\hat{\ddot{z}}_n \tag{22}$$

being Δt the time-step and,

$$\hat{\dot{z}}_n=-\left(\frac{2}{\Delta t}z_n+\dot{z}_n\right) \tag{23}$$

$$\hat{\ddot{z}}_n=-\left(\frac{4}{\Delta t^2}z_n+\frac{4}{\Delta t}\dot{z}_n+\ddot{z}_n\right) \tag{24}$$

Dynamic equilibrium can be established at time step n+1 by introducing the difference equations (21) and (22) into the equations of motion (1), leading to a non-linear system of algebraic equations, where the z_{n+1} are the unknowns,

$$f\left(z_{n+1}\right)=0 \tag{25}$$

Such system can be solved by the Newton-Raphson iteration, where the approximated tangent matrix is:

$$f_z=M+\frac{\Delta t}{2}C+\frac{\Delta t^2}{4}\left(\Phi_z^t\alpha\Phi_z+K\right) \tag{26}$$

and the residual vector:

$$f=\frac{\Delta t^2}{4}\left(M\ddot{z}+\Phi_z^t\alpha\Phi+\Phi_z^t\lambda^*-Q\right) \tag{27}$$

where C and K represent the contribution of damping and elastic forces of the system provided they exist. The calculation of these terms is explained below.

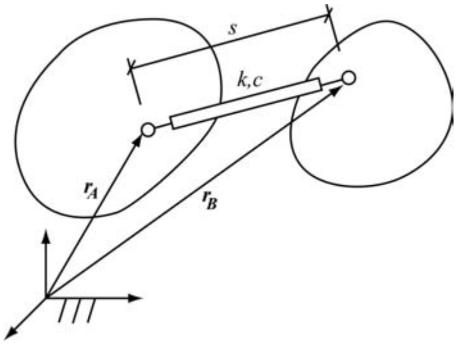

Figure 2. Matrices *K*, *C* due to springs and dampers.

Figure 2 shows a spring or damper connecting two bodies of a generic mechanism. In the case of a spring, if *f* is the force it exerts, the spring contribution to the stiffness matrix ***K*** can be expressed as,

$$\boldsymbol{K} = -\boldsymbol{Q}_z = -\left(\boldsymbol{s}_z^{\mathrm{t}} f\right)_z = -\boldsymbol{s}_z^{\mathrm{t}} \boldsymbol{f}_z = -\boldsymbol{s}_z^{\mathrm{t}} f_s \boldsymbol{s}_z \tag{28}$$

where the second derivative of the distance *s* with respect to ***z*** twice has been neglected. The elements of vector $\boldsymbol{s}_z$ correspond to the time-derivatives of distance *s* when unit velocities are successively given to the relative coordinates ***z***,

$$s_{\boldsymbol{z}_j} = \dot{s}\left(\dot{z}_j = 1, \dot{z}_i = 0, i \neq j\right) = \boldsymbol{u}_{\boldsymbol{AB}}^{\mathrm{t}}\left(\dot{\boldsymbol{r}}_B - \dot{\boldsymbol{r}}_A\right)\left(\dot{z}_j = 1, \dot{z}_i = 0, i \neq j\right) \tag{29}$$

being $\boldsymbol{u}_{AB}$ the unit vector pointing from *A* towards *B*.

If the spring is linear, with coefficient *k*, the form of the force *f* is,

$$f = -k\left(s - s_0\right) \tag{30}$$

being s_0 the natural length of the spring. Then, the derivative of the force *f* with respect to the distance *s* is given by $f_s = -k$, and Eq. (28) becomes,

$$\boldsymbol{K} = \boldsymbol{s}_z^{\mathrm{t}} k \boldsymbol{s}_z \tag{31}$$

It should be pointed out that only the components of $\boldsymbol{s}_z$ corresponding to the relatives coordinates ***z*** whose variation produces a change in the spring length *s*, have to be calculated, since the others are zero. Therefore, based on the topology of the mechanism, the decision of which elements of $\boldsymbol{s}_z$ must be obtained, can easily be taken.

In the case of a damper, if *f* is the force it exerts, the damper contribution to the damping matrix can be expressed as,

$$\boldsymbol{C} = -\boldsymbol{Q}_{\dot{z}} = -\left(\boldsymbol{s}_z^{\mathrm{t}} f\right)_{\dot{z}} = -\boldsymbol{s}_z^{\mathrm{t}} \boldsymbol{f}_{\dot{z}} = -\boldsymbol{s}_z^{\mathrm{t}} f_{\dot{s}} \dot{\boldsymbol{s}}_{\dot{z}} = -\boldsymbol{s}_z^{\mathrm{t}} f_{\dot{s}} \boldsymbol{s}_z \tag{32}$$

where the second derivative of the distance *s* with respect to ***z*** and $\dot{\boldsymbol{z}}$ has been neglected, and the identity $\dot{\boldsymbol{s}}_{\dot{z}} = \boldsymbol{s}_z$ has been applied.

If the damper is linear, with coefficient *c*, the form of the force *f* is,

$$f = -c\dot{s} \tag{33}$$

whose derivative with respect to the time-derivative of distance *s* is given by $f_{\dot{s}} = -c$, so that Eq. (32) becomes,

$$\boldsymbol{C} = \boldsymbol{s}_z^{\mathrm{t}} c \boldsymbol{s}_z \tag{34}$$

The procedure explained above, based on Eq. (26) and (27), yields a set of positions $\boldsymbol{z}_{n+1}$ that not only satisfies the equations of motion (1), but also the constraint conditions $\boldsymbol{\Phi} = 0$. However, it is not expected that the corresponding sets of velocities and accelerations satisfy $\dot{\boldsymbol{\Phi}} = 0$ and $\ddot{\boldsymbol{\Phi}} = 0$, because these conditions have not been imposed in the solution process. To overcome this difficulty, mass-damping-stiffness-orthogonal projections in velocities and accelerations are performed. It can be seen that the projections leading matrix is the same tangent matrix appearing in Eq. (26). Therefore, triangularization is avoided and projections in velocities and accelerations are carried out with just forward reductions and back substitutions.

If $\dot{\boldsymbol{z}}^*$ and $\ddot{\boldsymbol{z}}^*$ are the velocities and accelerations obtained after convergence has been achieved in the Newton-Raphson iteration, their cleaned counterparts $\dot{\boldsymbol{z}}$ and $\ddot{\boldsymbol{z}}$ are calculated from,

$$\boldsymbol{f}_z \dot{\boldsymbol{z}} = \left[\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K}\right]\dot{\boldsymbol{z}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_z^t \alpha \boldsymbol{\Phi}_t \tag{35}$$

for the velocities, and,

$$\boldsymbol{f}_z \ddot{\boldsymbol{z}} = \left[\boldsymbol{M} + \frac{\Delta t}{2}\boldsymbol{C} + \frac{\Delta t^2}{4}\boldsymbol{K}\right]\ddot{\boldsymbol{z}}^* - \frac{\Delta t^2}{4}\boldsymbol{\Phi}_z^t \alpha\left(\dot{\boldsymbol{\Phi}}_z \dot{\boldsymbol{z}} + \dot{\boldsymbol{\Phi}}_t\right) \tag{36}$$

for the accelerations.

4. EXAMPLES

To show the advantages of the new proposed formulation (called hybrid), the following four examples have been solved through three approaches: the new formulation and the two starting ones (called global and topological, respectively). The implementation of the three first cases has been carried out in the computing environment Matlab, which means that the reported CPU times would be drastically reduced if the same programs were written in Fortran or C languages. However, in order to show the behavior of the methods when facing a complex and realistic problem, and when all the resources available are used to achieve real-time performance, the fourth case has been implemented in Fortran language.

4.1. PLANAR MECHANISM

The first example is a one degree-of-freedom assembly of two four-bar linkages, illustrated in Fig. 3, which has been proposed [5] as an example to test the performance in cases where the mechanism undergoes singular configurations. When the mechanism reaches a horizontal position, the number of degrees of freedom instantaneously increases from 1 to 3. All the links have a uniformly distributed unit mass and a unit length. The gravity force acts in the negative vertical direction. At the beginning, the links pinned to the ground are in vertical position and upwards, receiving a unit clockwise angular velocity.

Figure 3. Double four-bar mechanism.

The simulation lasts for 10 s, during which the mechanism goes through the singular position ten times. Table 1 shows the obtained results by using the three methods for different time-steps (Δt). The error is measured as the maximum deviation, in J, suffered by the system energy. The time is the CPU time, in s, required to carry out the simulation.

	Global		Topological		Hybrid	
Δt	Error	Time	Error	Time	Error	Time
0.01	-0.21	0.86	no convergent		-0.01	3.20
0.03	-0.83	0.37	no convergent		-0.12	1.32
0.05	-9.03	0.26	no convergent		-0.34	0.83
0.1	wrong results		no convergent		-1.34	0.52

Table 1. Results for the double four-bar mechanism.

First of all, it can be seen that the topological method cannot accomplish the integration. It is due to the second velocity projection, which cannot be performed at the singularity. It can also be observed that the global method is more efficient than the hybrid one for the same time-step, while the hybrid method is found to be more accurate and robust, as it shows an smaller error for the same time-step and, moreover, works for larger time-steps than the global one. Therefore, if the efficiency of both methods is compared for the same level of error, similar performances would be obtained. It should be noted that, in this example, the number of global coordinates is 6, while the number of relative coordinates is 5. It is not then a case in which the hybrid method can get a great advantage due to the reduced number of coordinates with respect to the global method (this will occur in large examples).

4.2. CAR SUSPENSION

The second example, shown in Fig. 4, is the suspension system of the Iltis vehicle [6], which was used as a benchmark problem by the European automobile industry to check multibody dynamic codes.

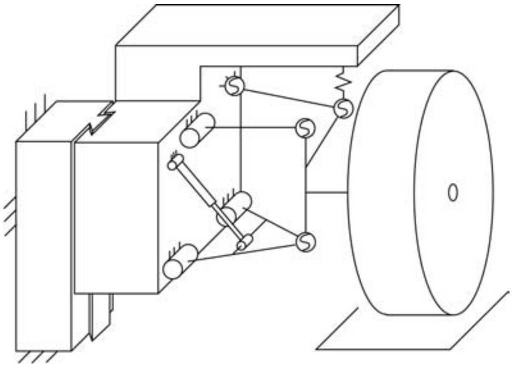

Figure 4. Iltis suspension.

The suspension starts moving from at rest conditions, and its position does not correspond to the static equilibrium. Thus, it freely oscillates until the static equilibrium position is reached. After three seconds, the suspension drops down a kerb of 10 cm height, and afterwards oscillates until the equilibrium is reached again. The complete analysis lasts for 7 s. It is carried out using the three methods, the performance results being depicted in Table 2. This time the error is obtained as the difference, in J, between the lost gravitational energy and the energy dissipated by the shock-absorber.

	Global		Topological		Hybrid	
Δt	Error	Time	Error	Time	Error	Time
0.001	-43.13	17.8	-43.49	45.7	-42.05	43.2
0.01	-43.45	2.04	-44.06	6.04	-42.80	5.44
0.03	-57.96	0.84	no convergent		no convergent	

Table 2. Results for the Iltis suspension.

In this case, unlike the previous one, the system shows a stiff behaviour, since the block representing one fourth of the whole vehicle has great inertia and, therefore, moves at slow frequency, while the wheel possesses little mass and, subsequently, moves at high frequency. Firstly, it can be seen that the topological method is the least efficient and robust. On the other hand, the global method is less accurate but more robust than the hybrid one, since it works for larger time-steps. Regarding efficiency, the global method keeps the same advantage with respect to the hybrid one than in the previous example, since 25 global coordinates have been used face to 8 relative coordinates. It is supossed that many more variables are needed, so that the global method requires a huge effort to solve the equation system, thus balancing its advantage.

4.3. SERIAL ROBOT

The PUMA robot, designed by Unimation & Co. and shown in Fig. 5, is an example of a 6 degrees-of-freedom serial manipulator. It has been often used by different authors [7] to illustrate methods and procedures in several areas of robotics. In this work, the robot has been taken as an example of multibody system undergoing changing configurations.

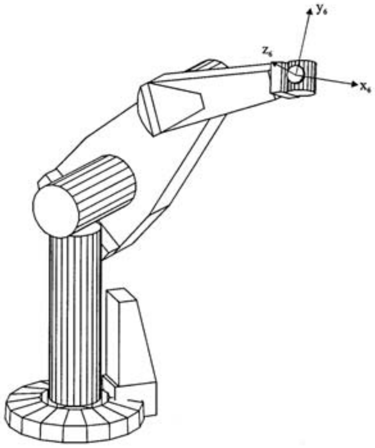

Figure 5. PUMA robot.

Starting from rest, torques at the six hinges of the robot are provided so that, in a time of 2 s, it arrives at a new position in the space, again in rest conditions. Once the new position has been reached, a point of the hand is attached to the ground, so that the robot loses 3 degrees-of-freedom. In this new configuration, torques are applied to the three rotational pairs of the hand, and a second manoeuvre, which lasts 4 s and ends with rest conditions, is performed. Finally, the robot is released from its attachment and returned in 2 s to the initial position by torques acting at the six revolute joints, once more finishing the manoeuvre at rest conditions. Therefore, the total simulation time is 8 s. Table 3 illustrates the results obtained when applying the three methods, where the error has been calculated as the distance, in mm, between the hand positions at initial and final times (which, ideally, should be coincident).

	Global		Topological		Hybrid	
Δt	Error	Time	Error	Time	Error	Time
0.01	0.53	6.25	0.26	3.05	0.34	4.02
0.05	3.4	1.92	1.5	1.20	1.8	1.74
0.1	no convergent		3.2	0.78	no convergent	

Table 3. Results for the PUMA robot.

This time, the global method shows the worst behaviour either in efficiency, accuracy or robustness, probably because the number of global coordinates defined to model the robot is 45 (sparse matrix techniques have not been used in this case), far from the 6 corresponding relative coordinates needed. Comparing the topological and the hybrid methods, some advantage is attained by the former with respect to the latter. It must be pointed out that the integration scheme used for the three methods is a single-step one. This fact favours the topological method, since every time the configuration changes, also does the number of integrable variables when using that method (it is not the case with the two others). If a multi-step integrator had been used, a considerable extra-effort would have been needed when applying the topological method, in order to restart the integration whenever a change in the configuration had place. Moreover, during the closed-chain topology phase of the robot motion, the topological method ought to verify, at each time, the validity of the degrees-of-freedom selected (i.e. the rank of the second velocity projection Jacobian), which has not actually been implemented since the correct degrees-of-freedom for such phase of the motion have been chosen.

4.4. ILTIS VEHICLE

The fourth example, shown in Fig. 6, is the full model of the Iltis vehicle [6], already referenced when describing the second example.

The simulation consists of 8 s of motion with the vehicle going up an inclined ramp and then down a series of stairs, at a constant horizontal speed of 5 m/s (the road profile is illustrated in Fig. 7). The simulation leads to a rather violent motion with acceleration peaks of up 5*g*.

As said above, this time the programs have been implemented in Fortran language and, furthermore, sparse matrix technologies have been applied where needed, so that the potential of the proposed method for achieving real-time performance when dealing with complex and realistic models can be perceived. The simulation has been carried out by means of the three methods studied so far in this paper, and a topological fully-recursive one [8], based in the articulated-inertia method [9], which proved to be very efficient although very difficult to implement in previous studies [10]. Moreover, in order to have a better reference to appraise the attained efficiency, commercial software ADAMS has also been used to perform the simulation. It should be pointed out that ADAMS default options [11] have been chosen to run the simulation: variable time-step size Gear integrator along with an index-3 formulation, error of 10^{-3} and Jacobian evaluation in one out of four iterations. All the programs have been run on a 1,200 € PC with one AMD Athlon XP processor 1600+ @ 1.4 GHz.

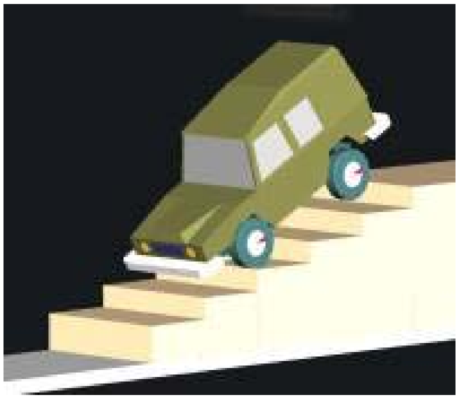

Figure 6. Iltis vehicle.

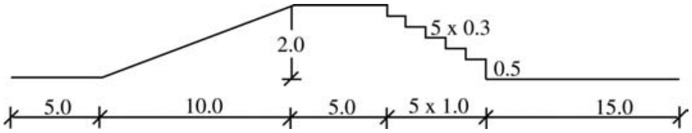

Figure 7. Road profile.

Table 4 illustrates the CPU times, in s, obtained when applying the four methods and the ADAMS solver. Given that the four methods being compared use a fix time-step size integrator scheme while ADAMS features a variable time-step size procedure, the following criterium has been adopted in order to present the results: first, a fix time-step size of 0.01 s has been selected for the four methods and, later on, the largest possible fix time-step size for which good results are obtained with each method has been also considered, so as to show not only efficiency but robustness as well. The fully-recursive formulation constitutes an exception, as it achieves the best performance with a time-step size smaller than 0.01 s. The reason is that such method employs a fixed point iteration scheme for the integration process, so that large time-step sizes ask for a high number of iterations to converge. Finally, the last column of Table 4 shows how many times the method is faster than real-time, which gives simple and clear information about the method performance.

Global	Δt=0.01	5.41	2.11
	Δt=0.0175	3.80	
Topological (semi-recurs.)	Δt=0.01	0.74	17.02
	Δt=0.025	0.47	
Hybrid	Δt=0.01	0.72	21.05
	Δt=0.035	0.38	
Topological (fully-recurs.)	Δt=0.01	3.06	3.48
	Δt=0.0075	2.30	
ADAMS	Δt variable	7.62	1.05

Table 4. Results for the Iltis vehicle.

In this example, it turns out that, definitely, the global method is largely the least efficient among the three being compared. This can be explained by the fact that the global method needs 168 variables to model the vehicle, while the number of relative coordinates required by both the topological and hybrid methods is only 26 (from which 10 are the system degrees-of-freedom). Moreover, the global method is also the least robust, since the largest time-step size it can reach while giving correct results is clearly smaller than that of its competitors.

Regarding the comparison between the topological semi-recursive and hybrid methods, a substantial advantage of the hybrid formulation is observed, reaching higher levels of efficiency due to a greater robustness. Once again, the detection procedure for the validity of the selected degrees-of-freedom always needed by the topological method at each time (as it integrates independent variables), has not been implemented, so that the advantage in favour of the hybrid method should be even larger.

The topological fully-recursive method, included for this last example only due to the high efficiency it had shown in previous works [8], is far behind the semi-recursive and hybrid formulations. Besides, taking into account that implementation of such method is very involved and case-dependent, it should be discarded as a competitor of the mentioned alternatives, more efficient and easy-to-implement.

Finally, comparison with commercial software ADAMS shows that the proposed formulation can be seen as a good candidate for real-time general-purpose applications, as it encompasses good levels of efficiency, accuracy and robustness along with a reasonable easiness of implementation.

5. CONCLUSIONS

Based on the results obtained for the studied examples, the conclusions can be drawn as follows:

(1) A new real-time formulation for the dynamics of multibody systems has been presented, which encompasses high ranks of efficiency, accuracy, robustness and easiness of implementation.

(2) The method combines a topological semi-recursive formulation based on velocities projection and a global penalty formulation for closed-loops consideration.

(3) The method is more robust that its topological predecessor as: a) it can handle singular positions, since the penalty approach produces a positive-definite leading matrix under any conditions; b) it does not need to check the validity of the degrees-of-freedom at each time-step, in order to change the set of variables to be integrated in case they become dependent. This second aspect will be particularly advantageous when dealing with changing configurations (which may appear due to dry friction, joint backlash, unilateral contacts, grasping actions, etc.).

(4) The method is more efficient that its global ancestor for a similar level of accuracy when the number of global coordinates becomes large. In cases of complex and realistic systems (e.g. the whole model of a vehicle), this advantage can reach one order of magnitude.

(5) The method shows an excellent level of efficiency with respect to commercial software. Therefore, it is foreseen as a good candidate for general-purpose real-time applications.

ACKNOWLEDGMENTS

This research has been sponsored by the Spanish CICYT (Grant No. DPI2000-0379) and the Galician SGID (Grant No. PGIDT01PXI16601PN).

REFERENCES

[1] Shabana A.A., 1998, *Dynamics of multibody systems*, Cambridge University Press, Cambridge.

[2] García de Jalón J., and Bayo E., 1994, *Kinematic and dynamic simulation of multibody systems*, Springer-Verlag, New York.

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

[4] Rodríguez J.I., 2000, "Analisis eficiente de mecanismos 3D con metodos topologicos y tecnologia de componentes en Internet," Ph. D. Thesis, University of Navarre, San Sebastian, Spain.

[5] Bayo E., and Avello A., 1994, "Singularity free augmented Lagrangian algorithms for constraint multibody dynamics," Nonlinear Dynamics, **5**, pp. 209-231.

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

[7] Angeles J., and Ma O., 1988, "Dynamic simulation of n-axis serial robotic manipulators using a natural orthogonal complement," Int. J. of Robotic Research, **7**, pp. 32-47.

[8] Jiménez J.M., 1993, "Kinematic and dynamic formulations for real-time simulation of multibody systems," Ph. D. Thesis, University of Navarre, San Sebastian, Spain.

[9] Featherstone R., 1987, *Robot dynamics algorithms*, Kluwer Academic Publishers, Dordrecht.

[10] Cuadrado J., Cardenal J., Morer, P., and Bayo, E., 2000, "Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments," Multibody System Dynamics, **4**, pp. 55-73.

[11] ADAMS 11.0.0, 2000, *Product Guides*.