MULTIBODY DYNAMICS 2003
Jorge A.C. Ambrósio(Ed.)
IDMEC/IST, Lisbon, Portugal, July 1-4 2003

PENALTY, SEMI-RECURSIVE AND HYBRID METHODS FOR MBS REAL-TIME DYNAMICS IN THE CONTEXT OF STRUCTURAL INTEGRATORS

Javier Cuadrado, Daniel Dopico

Escuela Politécnica Superior
Universidad de La Coruña
Mendizábal s/n, 15403 Ferrol, Spain
e-mail: javicuad@cdf.udc.es, web page: http://lim.ii.udc.es

Keywords: Real-Time Dynamics, Penalty Formulations, Semi-recursive Formulations, Global Methods, Topological Methods, Structural Integrators

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 (penalty) and one of the most systematic topological formulations (semi-recursive). 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, a rather exigent simulation of the full model of a car vehicle is carried out using the three formulations and a commercial tool, so as to provide the readers with a well-known reference for comparison. Furthermore, the example is also analyzed through the three dynamic formulations in combination with different structural integrators, so that the influence of the integration scheme on the method performance can be appraised.*

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 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, or optimization procedures for design purposes carried out, 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/or kinematic pair. Consequently, this family of methods leads to simple and general algorithms of easy implementation but not very efficient, as they produce models with large number of variables and large frequency dispersion. The use of sparse matrix techniques can contribute to alleviate the low efficiency of this family of methods.

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. Moreover, these methods produce models with a lower number of variables and more balanced frequencies than their global counterparts. However, these kinds 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, 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 in efficiency 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 $\mathbf{z}$ are the relative coordinates, $\mathbf{M}$ is the mass matrix of the mechanism expressed in terms of the relative coordinates, $\mathbf{\Phi}$ is the constraints vector due to the closure conditions of the loops, $\mathbf{\Phi}_{\mathbf{z}}$ is the Jacobian matrix of the constraints, α is the penalty factor, $\mathbf{Q}$ 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):

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

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 $\mathbf{M}$ and $\mathbf{Q}$, a second set of coordinates is defined. It can be expressed at velocity level for each body of the system in the form,

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

being $\dot{\mathbf{s}}$ 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],

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

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

wherein m is the body mass, $\mathbf{I}_3$ is the 3x3 identity matrix, $\mathbf{g}$ is the global position of the mass center of the body, $\tilde{\mathbf{g}}$ is the dual anti-symmetric matrix of $\mathbf{g}$, $\mathbf{J}$ 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, $\mathbf{f}$ is the vector of forces applied to the body, and $\mathbf{n}$ is the vector of applied moments with respect to the mass center of the body.

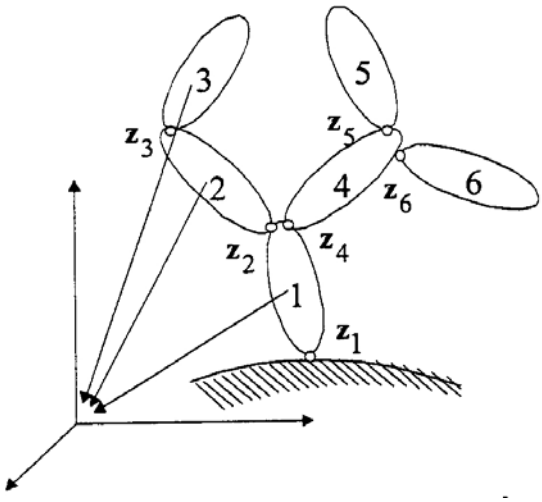

Figure 1: Example of mechanism topology.

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

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

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

$$\mathbf{R} = \mathbf{T}\mathbf{H} \tag{7}$$

with

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

and,

$$\mathbf{H}=\begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_4 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_5 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_6 \end{bmatrix} \tag{9}$$

where $\mathbf{I}_6$ is the 6x6 identity matrix and $\mathbf{R}_i$ (i=1,6) are 6x1 vectors.

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

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

where the superindex (*) is used for virtual velocities. Substituting in Eq. (10) the result of Eq. (6) together with its derivative yields

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

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

$$\mathbf{M}=\mathbf{R}^{t}\bar{\mathbf{M}}\mathbf{R} \tag{12}$$

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

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

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

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

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

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,

$$\mathbf{T}^t\mathbf{M}\mathbf{T} = \begin{bmatrix} \mathbf{M_1} & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{M}_4 & \mathbf{M}_5 & \mathbf{M}_6 \\ & \mathbf{M}_2 & \mathbf{M}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & \mathbf{M}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & \mathbf{M}_4 & \mathbf{M}_5 & \mathbf{M}_6 \\ & \textit{sym} & & & \mathbf{M}_5 & \mathbf{0} \\ & & & & & \mathbf{M}_6 \end{bmatrix} \tag{16}$$

where the sub-matrices are obtained as,

$$\begin{aligned} \mathbf{M}_6 &= \bar{\mathbf{M}}_6 \\ \mathbf{M}_5 &= \bar{\mathbf{M}}_5 \\ \mathbf{M}_3 &= \bar{\mathbf{M}}_3 \\ \mathbf{M}_2 &= \bar{\mathbf{M}}_2 + \mathbf{M}_3 \\ \mathbf{M}_4 &= \bar{\mathbf{M}}_4 + \mathbf{M}_5 + \mathbf{M}_6 \\ \mathbf{M}_1 &= \bar{\mathbf{M}}_1 + \mathbf{M}_2 + \mathbf{M}_4 \end{aligned} \tag{17}$$

and,

$$\mathbf{T}^t\left(\bar{\mathbf{Q}} - \bar{\mathbf{M}}\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right) = \begin{Bmatrix} \mathbf{Q_1} \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \\ \mathbf{Q}_4 \\ \mathbf{Q}_5 \\ \mathbf{Q}_6 \end{Bmatrix} \tag{18}$$

where the sub-vectors are obtained as,

$$\begin{aligned} \mathbf{Q}_6 &= \bar{\mathbf{Q}}_6 - \bar{\mathbf{M}}_6\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_6 \\ \mathbf{Q}_5 &= \bar{\mathbf{Q}}_5 - \bar{\mathbf{M}}_5\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_5 \\ \mathbf{Q}_4 &= \bar{\mathbf{Q}}_4 - \bar{\mathbf{M}}_4\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_4 + \mathbf{Q}_5 + \mathbf{Q}_6 \\ \mathbf{Q}_3 &= \bar{\mathbf{Q}}_3 - \bar{\mathbf{M}}_3\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_3 \\ \mathbf{Q}_2 &= \bar{\mathbf{Q}}_2 - \bar{\mathbf{M}}_2\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_2 + \mathbf{Q}_3 \\ \mathbf{Q}_1 &= \bar{\mathbf{Q}}_1 - \bar{\mathbf{M}}_1\left(\mathbf{T}\dot{\mathbf{H}}\dot{\mathbf{z}}\right)_1 + \mathbf{Q}_2 + \mathbf{Q}_4 \end{aligned} \tag{19}$$

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 $\mathbf{\Phi}_\mathbf{z}$, which implies the differentiation of the constraints vector with respect to the relative coordinates $\boldsymbol{z}$. This can be easily done by applying the chain differentiation rule as,

$$\mathbf{\Phi}_\mathbf{z} = \mathbf{\Phi}_\mathbf{q}\mathbf{q}_\mathbf{z} \tag{20}$$

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 $\mathbf{\Phi}_\mathbf{q}$ is the traditional Jacobian matrix of the constraints, when natural coordinates are used[2], while the term $\mathbf{q}_\mathbf{z}$ 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:

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

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

being Δt the time-step and,

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

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

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 $\mathbf{z}_{n+1}$ are the unknowns,

$$\mathbf{h}\left(\mathbf{z}_{n+1}\right) = 0 \tag{25}$$

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

$$\mathbf{h}_\mathbf{z} = \mathbf{M} + \frac{\Delta t}{2}\mathbf{C} + \frac{\Delta t^2}{4}\left(\mathbf{\Phi}_z^t \alpha \mathbf{\Phi}_z + \mathbf{K}\right) \tag{26}$$

and the residual vector:

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

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.

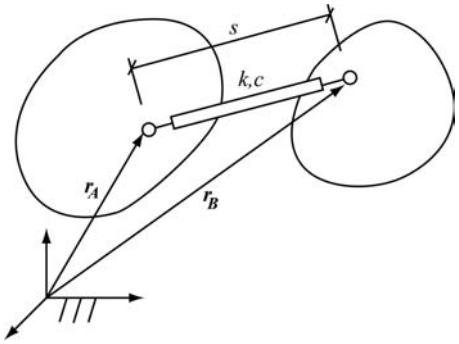

Figure 2: Matrices **K**, **C**, due to springs and dampers.

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,

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

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

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

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

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

$$f = -k\left(s - s_\mathrm{o}\right) \tag{30}$$

being s_o 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,

$$\mathbf{K} = \mathbf{s}_\mathbf{z}^\mathrm{t} k \mathbf{s}_\mathbf{z} \tag{31}$$

It should be pointed out that only the components of $\mathbf{s}_\mathbf{z}$ corresponding to the relative 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 $\mathbf{s}_\mathbf{z}$ 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,

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

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

$$\mathbf{C} = \mathbf{s}_\mathbf{z}^\mathrm{t} c\mathbf{s}_\mathbf{z} \tag{34}$$

The procedure explained above, based on Eq. (26) and (27), yields a set of positions $\mathbf{z}_{n+1}$ 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 $\dot{\mathbf{z}}^*$ and $\ddot{\mathbf{z}}^*$ are the velocities and accelerations obtained after convergence has been achieved in the Newton-Raphson iteration, their cleaned counterparts $\dot{\mathbf{z}}$ and $\ddot{\mathbf{z}}$ are calculated from,

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

for the velocities, and,

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

for the accelerations.

4 EXAMPLE: THE ILTIS VEHICLE

To show the advantages of the new proposed formulation (called hybrid) when facing a complex and realistic problem, the full model of the Iltis vehicle[5] (Fig. 3), which has been used as a benchmark problem by the European automobile industry to check multibody dynamic codes, has been solved through three approaches: the new formulation and the two starting ones (called global and topological, respectively).

The simulation consists of 8 s of motion with the vehicle going up an inclined ramp and then down a series of stairs, starting at a speed of 5 m/s (the road profile is illustrated in Fig. 3). A rather violent motion is undergone by the vehicle, with acceleration peaks of up 5*g*.

The programs have been implemented in Fortran language and, furthermore, sparse matrix technologies have been applied where needed, so that the potential of the three formulations being compared 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 described so far in this paper, and a topological fully-recursive one[6], based on the articulated-inertia method[7], which proved to be very efficient although very difficult to implement in previous studies[8]. 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[9] 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.

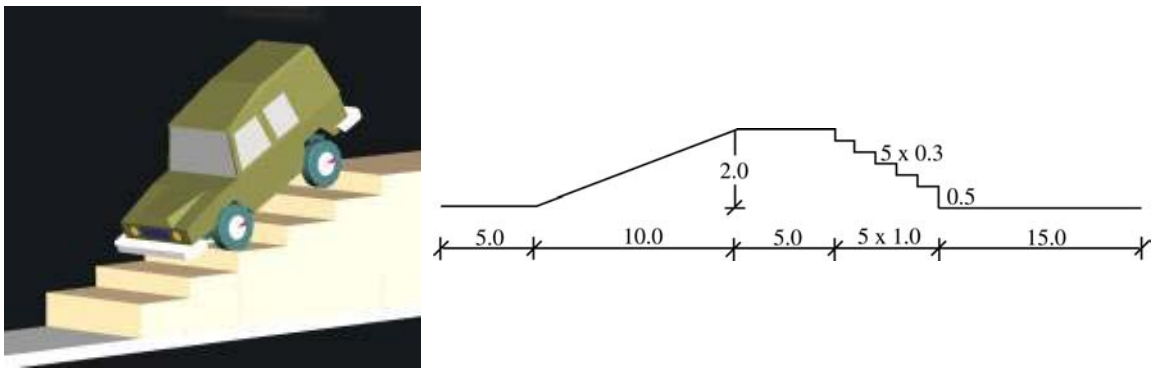

Figure 3: Iltis vehicle and road profile.

Table 1 illustrates the CPU times 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 1 shows how many times the method is faster than real-time, which gives simple and clear information about the method performance.

	Δt (s)	CPU time (s)	$\frac{\text{Real-time}}{\text{CPU time}}$
Global	0.01	5.41	2.07
	0.0175	3.86	
Topological (semi-recursive)	0.01	0.74	14.29
	0.025	0.56	
Hybrid	0.01	0.72	21.62
	0.035	0.37	
Topological (fully-recursive)	0.01	3.06	3.48
	0.0075	2.30	
ADAMS	variable	7.62	1.05

Table 1: Results for the Iltis vehicle.

In this example, it turns out that the global method is largely the least efficient among the main 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. 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 favor of the hybrid method should be even larger.

The topological fully-recursive method, included 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 the 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 INFLUENCE OF THE NUMERICAL INTEGRATOR

So far, the three dynamic formulations being compared (global, topological and hybrid) have been built using the trapezoidal rule as numerical integrator. In this section, the influence of the numerical integrator in the behavior of each method is to be addressed.

Three families of numerical integrators, all of them belonging to the so-called structural integrators group, are to be tested: the Newmark family with numerical dissipation, the HHT algorithm, and the Generalized-α method. The reason to use the structural integrators is that they show a completely analogous scheme to the one of the trapezoidal rule (which in fact is a particular case of all of them), so that the general strategy of combining the dynamic and integrator equations can be preserved.

The general form of the dynamic equations is,

$$\mathbf{M}(\mathbf{q})\cdot\ddot{\mathbf{q}}+\mathbf{P}(\mathbf{q},\dot{\mathbf{q}})=\mathbf{0} \tag{37}$$

where $\mathbf{M}(\mathbf{q})$ is the mass matrix and $\mathbf{P}(\mathbf{q},\dot{\mathbf{q}})$ contains all the terms which are constant or functions of positions and/or velocities. When the abovementioned integrators are applied, Eq. (37) turns into,

$$\left[\left(1-\delta_m\right)\cdot\mathbf{M}_{n+1}\cdot\ddot{\mathbf{q}}_{n+1}+\delta_m\cdot\mathbf{M}_n\cdot\ddot{\mathbf{q}}_n\right]+\left[\left(1-\delta_f\right)\cdot\mathbf{P}\left(\mathbf{q}_{n+1},\dot{\mathbf{q}}_{n+1}\right)+\delta_f\cdot\mathbf{P}\left(\mathbf{q}_n,\dot{\mathbf{q}}_n\right)\right]=\mathbf{0} \tag{38}$$

which means that the dynamic equilibrium is stated from partial contributions from step n and n+1.

On the other hand, the difference equations of Newmark integrator are:

$$\dot{\mathbf{q}}_{n+1}=\frac{\gamma}{\beta\cdot\Delta t}\cdot\mathbf{q}_{n+1}+\hat{\dot{\mathbf{q}}}_n \tag{39}$$

$$\ddot{\mathbf{q}}_{n+1}=\frac{1}{\beta\cdot\Delta t^2}\cdot\mathbf{q}_{n+1}+\hat{\ddot{\mathbf{q}}}_n \tag{40}$$

with,

$$\hat{\dot{\mathbf{q}}}_n=-\left(\frac{\gamma}{\beta\cdot\Delta t}\cdot\mathbf{q}_n+\left(\frac{\gamma}{\beta}-1\right)\cdot\dot{\mathbf{q}}_n+\Delta t\left(\frac{\gamma}{2\beta}-1\right)\ddot{\mathbf{q}}_n\right) \tag{41}$$

$$\hat{\ddot{\mathbf{q}}}_n=-\left(\frac{\gamma}{\beta\cdot\Delta t^2}\cdot\mathbf{q}_n+\frac{\gamma}{\beta\cdot\Delta t}\cdot\dot{\mathbf{q}}_n+\left(\frac{1}{2\beta}-1\right)\ddot{\mathbf{q}}_n\right) \tag{42}$$

Eqs. (38-42) are the basis from which the three families of integrators here considered are to be obtained. Four parameters appear in such equations: δ_m, δ_f, γ, β. Different values of these four parameters lead to the different families of integrators.

From the studies performed about structural integrators (usually for linear systems) concerning stability and accuracy issues[10], it is concluded that the more appropriate values for the parameters are those shown in Table 2. As it can be observed in Table 2, the four parameters of each family are obtained as functions of only one parameter, called in the Table the *basic parameter*. The admissible range of variation of such basic parameters is also illustrated in the Table.

	Newmark dissip.	HHT	Generalized-α
Basic parameter	$\xi \in [-1,0]$	$\delta_f \in \left[0,\frac{1}{3}\right]$	$\rho_\infty \in [0,1]$
δ_m	0	0	$\frac{2\rho_\infty - 1}{\rho_\infty + 1}$
δ_f	0	δ_f	$\frac{\rho_\infty}{\rho_\infty + 1}$
γ	$\frac{(1-2\xi)}{2}$	$\frac{(1+2\delta_f)}{2}$	$\frac{1}{2} - \delta_m + \delta_f$
β	$\frac{(1-\xi)^2}{4}$	$\frac{(1+\delta_f)^2}{4}$	$\frac{1}{4}(1+\delta_f-\delta_m)^2$

Table 2: Integrators parameters.

With this election of the parameters, the three families are unconditionally stable for linear systems. In what respects to the accuracy, the HHT and the Generalized-α algorithms are second order, while the Newmark dissipative is only first order accurate.

In order to find out the influence of the numerical integrator selected on the final performance of the three dynamic formulations previously compared in this paper, the three families of integrators (Newmark dissipative, HHT, Generalized-α) have been implemented for each dynamic formulation (global, topological, hybrid). Then, the already described simulation of the Iltis vehicle has been executed with each combination of dynamic formulation and numerical integrator (i.e. nine combinations are possible).

Multiple executions have been carried out for each combination, so as to perform a two-dimensional sweeping: time-step and basic parameter. Table 3 shows the ranges of variation of the basic parameter and the time-step for each integrator family, along with the increments considered for these magnitudes to perform the different executions.

	Newmark dissip.	HHT	Generalized-α
Basic parameter	$\xi \in [-1,0]$	$\delta_f \in \left[0,\frac{1}{3}\right]$	$\rho_\infty \in [0,1]$
Increment	0.02	0.01	0.02
Δt (s)	[0.01,0.05]	[0.01,0.05]	[0.01,0.05]
Increment (s)	0.0025	0.0025	0.0025

Table 3: Range of variation and increment of basic parameter and time-step for each integrator family.

For all the executions, the CPU time spent to perform the simulation has been recorded, as well as the solution error. To evaluate such error, the solution attained with the corresponding dynamic formulation and the trapezoidal rule for a time-step of 0.001 s has been taken as reference. Then, the error of a certain combination of dynamic formulation and integrator has been obtained as,

$$error = \frac{1}{3(ns+1)}\left[\sum_{i=1}^{3}\left(\sqrt{\sum_{j=0}^{ns}\left(z_{ij}-z_{ij}^{*}\right)^2}\right)\right] \tag{43}$$

where ns stands for the number of time-steps, z_{1j} represents the history of the vertical coordinate of a chassis point in the neighborhood of the mass center, z_{2j} and z_{3j} represent the histories of the vertical coordinates of the center of the front left and rear right wheel respectively, and z^* are their counterparts for the simulation taken as reference. Only executions producing an error under $5 \cdot 10^{-3}$ have been considered acceptable. In the following subsections, results of the study are described for each dynamic formulation.

5.1 Global formulation

Table 4 shows the most efficient integration scheme for each integrator family combined with the global dynamic formulation. The most efficient results when using the trapezoidal rule are also given for comparison. In the Table, the fastest method has been bolfaced.

	Basic parameter	Δt (s)	CPU time (s)	$\frac{\text{Real-time}}{\text{CPU time}}$	Error ($\times 10^{-3}$)
Trapez. rule	-	0.0175	3.86	2.07	1.536
Newmark dis.	**-0.62**	**0.045**	**1.74**	**4.60**	**3.994**
HHT	0.15	0.03	2.62	3.05	2.468
Generalized-α	0	0.025	2.77	2.89	2.428

Table 4: The most efficient integration schemes for the global dynamic formulation.

The trend of Newmark dissipative when combined with the global formulation is that bigger damping produces better convergence but worse accuracy. However, this family of integrators enables to use a larger time-step than the trapezoidal rule, while keeping a good level of accuracy, thus achieving high efficiency rates.

Regarding the HHT algorithm, when small time-steps are used, the trapezoidal rule shows to be more efficient and accurate. However, for large time-steps the trend is just the opposite.

In what respects to the Generalized-α method, the trapezoidal rule is clearly superior for small time-steps. However, the trend can be inverted for large time-steps if a good value of the spectral radius ρ_∞ is selected for the Generalized-α integrator. Such selection should be carried out very carefully, as the algorithm is very sensitive to small changes in the value of the spectral radius. A too high value of ρ_∞ leads to unstable behavior, while a too low value of ρ_∞ eliminates an excessive part of the solution.

5.2 Topological formulation

Table 5 shows the most efficient integration scheme for each integrator family combined with the topological dynamic formulation. The most efficient results when using the trapezoidal rule are also given for comparison. In the Table, the fastest method has been bolfaced.

	Basic parameter	Δt (s)	CPU time (s)	$\frac{\text{Real-time}}{\text{CPU time}}$	Error ($\text{x}10^{-3}$)
Trapez. rule	-	0.025	0.56	14.29	4.040
Newmark dis.	**-0.28**	**0.0225**	**0.55**	**14.55**	**1.717**
HHT	0.01	0.025	0.57	14.04	4.001
Generalized-α	0	0.0225	0.68	11.76	3.454

Table 5: The most efficient integration schemes for the topological dynamic formulation.

The trend of Newmark dissipative when combined with the topological formulation is more or less the same that the one observed with the global formulation: more damping allows for larger time-steps. However, this time the achieved improvement with respect to the trapezoidal rule is not relevant, as it was in the case of the global formulation.

The HHT algorithm follows a similar behavior to that of Newmark dissipative, showing no efficiency improvement over the trapezoidal rule.

Likewise, the Generalized-α method finds problems when combined with the topological method. It is not worth using it: the trapezoidal rule leads to better results for all the time-steps considered.

5.3 Hybrid formulation

Table 6 shows the most efficient integration scheme for each integrator family combined with the hybrid dynamic formulation. The most efficient results when using the trapezoidal rule are also given for comparison. In the Table, the fastest method has been bolfaced.

	Basic parameter	Δt (s)	CPU time (s)	$\frac{\text{Real-time}}{\text{CPU time}}$	Error ($\times 10^{-3}$)
Trapez. rule	-	0.035	0.37	21.62	4.508
Newmark dis.	**-0.52**	**0.05**	**0.25**	**32.00**	**4.117**
HHT	0.35	0.0425	0.33	24.24	2.450
Generalized-α	0	0.05	0.28	28.57	2.834

Table 6: The most efficient integration schemes for the hybrid dynamic formulation.

The trend of Newmark dissipative, this time combined with the hybrid formulation, keeps the same that the one observed with the global formulation. A moderate value of damping enables to take larger time-steps than the trapezoidal rule, while preserving the accuracy, so leading to more efficient executions.

Regarding the HHT algorithm, its behavior also mimics the one observed with the global formulation. When small time-steps are used, the trapezoidal rule shows to be more efficient and accurate. However, for large time-steps the trend is just the opposite and good levels of efficiency are attained.

In what respects to the Generalized-α method, the same behavior is observed that in the case of the global formulation. The trapezoidal rule is superior for small time-steps, but the trend can be inverted for large time-steps if the adequate value of the spectral radius is selected for the Generalized-α integrator.

5.4 General comparison

Table 7 summarizes the fastest methods obtained with each dynamic formulation, listed by order of efficiency.

	Δt (s)	CPU time (s)	$\frac{\text{Real-time}}{\text{CPU time}}$
Hybrid + Newmark dissip.	0.05	0.25	32.00
Topological + Newmark dissip.	0.0225	0.55	14.55
Global + Newmark dissip.	0.045	1.74	4.60

Table 7: The most efficient methods.

As shown in Table 7, the advantage in efficiency of the hybrid formulation over the topological one is increased over the case in which the trapezoidal rule is used as numerical integrator (see Table 1). The reason is that the hybrid formulation clearly benefits from the use of structural integrators with some numerical damping, while the topological formulation does not improve its results under the same circumstances. As the hybrid formulation, the global one also gets a great benefit from the combination with dissipative integrators. However, it does not reach the levels of efficiency attained by the topological method.

For the reader to appraise the accuracy obtained in the fastest simulation presented, a comparison between such solution and that used as reference to evaluate the error (trapezoidal rule at time-step of 0.001 s) is illustrated in Fig. 4.

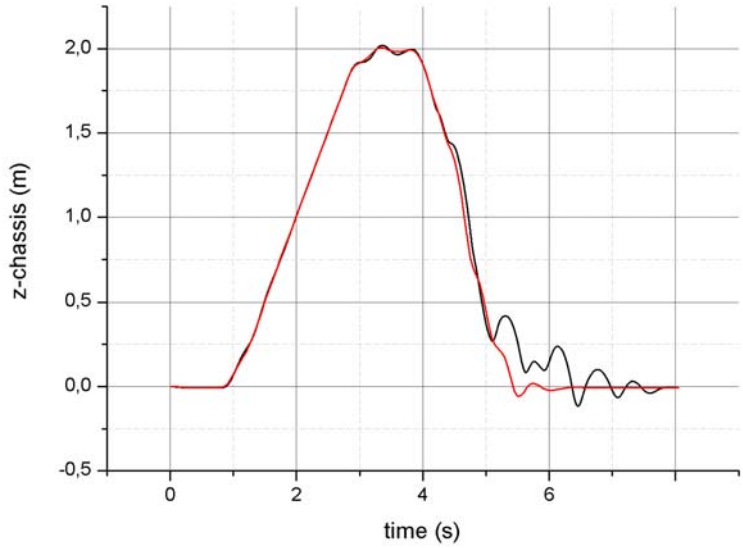

Figure 4: The fastest simulation vs. the reference simulation.

It can be seen in Fig. 4 that significant differences are obtained during the final and more violent phase of the motion. The reason is that the method achieves its excellent level of stability, which in turn allows for incredibly large time-steps and very high efficiency rates, at the price of introducing some numerical damping, thus suppressing part of the solution. Moreover, since so large time-steps (0.05 s) are used, many things happen between two consecutive time-steps, which also has the result of losing information about the system response. Therefore, depending on the application, the analyst will have to accept a compromise between the levels of accuracy and efficiency he desires to achieve.

6 CONCLUSIONS

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

a) 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.
b) The method combines a topological semi-recursive formulation based on velocities projection and a global penalty formulation for closed-loops consideration.
c) 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.).
d) 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.
e) 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.
f) When combined with structural integrators which introduce numerical damping, the method obtains great benefit, so notably improving its efficiency. The same trend is observed for the global method. However, the topological one does not benefit from such combination.

ACKNOWLEDGMENTS

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

REFERENCES

[1] A.A. Shabana, *Dynamics of multibody systems*, Cambridge University Press, Cambridge, United Kingdom (1998).
[2] J. García de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems*, Springer-Verlag, New York, U.S.A. (1994).
[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", *Int. J. for Numerical Methods in Engineering*, **51**, 1033-1052 (2001).

[4] J.I. Rodriguez, *Analisis eficiente de mecanismos 3D con metodos topologicos y tecnologia de componentes en Internet*, Ph. D. Thesis, University of Navarre, San Sebastian, Spain (2000).
[5] Iltis Data Package, *IAVSD Workshop*, Herbertov, Czechoslovakia (1990).
[6] J.M. Jimenez, *Kinematic and dynamic formulations for real-time simulation of multibody systems*, Ph. D. Thesis, University of Navarre, San Sebastian, Spain (1993).
[7] R. Featherstone, *Robot dynamics algorithms*, Kluwer Academic Publishers, Dordrecht, The Netherlands (1987).
[8] J. Cuadrado, J. Cardenal, P. Morer and E. Bayo, "Intelligent simulation of multibody dynamics: space-state and descriptor methods in sequential and parallel computing environments", *Multibody System Dynamics*, **4**, 55-73 (2000).
[9] ADAMS 11.0.0, *Product Guides* (2000).
[10] M. Geradin and A. Cardona, Flexible Multibody Dynamics, John Wiley & Sons, 2nd ed., Chichester, United Kingdom (2001).