Proceedings of MUSME 2005, the International Symposium
on Multibody Systems and Mechatronics
Uberlandia (Brazil), 6-9 March 2005
Paper n. 06-MUSME05

ON THE APTITUDE OF IRK INTEGRATORS FOR THE REAL-TIME DYNAMICS OF MULTIBODY SYSTEMS

D. Dopico, U. Lugris, M. Gonzalez, J. Cuadrado
Laboratory of Mechanical Engineering, University of La Coruña
Mendizabal s/n, 15403 Ferrol, Spain
Email: ddopico@mail2.udc.es; Web: http://lim.ii.udc.es

Keywords: multibody systems, dynamics, real-time simulation, numerical integration, IRK

ABSTRACT

During the last years, the authors have been conducting a study on the aptitude of different methods for real-time applications in the field of multibody dynamics.

In a first work, the authors presented a new formulation, called hybrid, obtained as combination of a topological semi-recursive formulation based on velocity transformations, and a global penalty formulation for closed-loops consideration. Using the trapezoidal rule as numerical integrator for the three formulations, they were compared, and the conclusion was that the proposed formulation was more robust and efficient that its predecessors for large problems.

In a second work, a new comparison among the three formulations was carried out, but this time the trapezoidal rule was substituted by different structural integrators: Newmark dissipative schemes, HHT rule, and the Generalized-α family. It was shown that, for large multibody systems, Newmark dissipative was the best election since, provided that the adequate parameters were chosen, excellent behavior was achieved in terms of efficiency, robustness and accuracy.

In the present paper, the performance of the three mentioned formulations in combination with another group of integrators, the Implicit Runge-Kutta family (IRK), is analyzed. The purpose is to clarify which kind of IRK algorithms can be more suitable for real-time applications, and to see whether they can be competitive with the already tested structural family of integrators. The final objective of the work is to provide some practical criteria for those interested in achieving real-time performance for large and complex multibody systems.

1. INTRODUCTION

Real-time performance in multibody dynamics is being more and more demanded for applications such as human- and hardware-in-the-loop simulations or intelligent vehicle control systems. Although faster processors are continuously provided by the computer industry, research on the corresponding mechanical and mathematical formulations is also needed, since users will always want to simulate larger and more complex systems, and to obtain more accurate results, which in turn asks for more detailed models.

With this aim, during the last years the authors have been conducting a study on the aptitude of different methods for real-time applications in the field of multibody dynamics.

In a first work [Cuadrado et al. 2004a], the authors presented a new real-time formulation for the dynamics of multibody systems, which encompasses high ranks of efficiency, accuracy, robustness and easiness of implementation. The new method, called hybrid, was obtained as combination of a topological semi-recursive formulation based on velocity transformations [Garcia de Jalon et al. 2003], and a global penalty formulation for closed-loops consideration [Cuadrado et al. 2001]. The three formulations (we will refer to them as global, topological and hybrid) were compared through the analysis of several multibody systems, and the proposed formulation showed to be more robust and efficient that its predecessors for large problems. The implicit, single-step trapezoidal rule was used as numerical integrator for the three formulations compared.

In a second work [Cuadrado et al. 2004b], a new comparison among the three formulations was carried

out. Given that Newmark-type integrators, widely used in structural dynamics, had shown to adapt well to the equations of motion of multibody systems, even for real-time purposes [Garcia de Jalon and Bayo 1994], [Geradin and Cardona 2001], the trapezoidal rule was substituted by different structural integrators: Newmark dissipative schemes, HHT rule, and the Generalized-α family. It was shown that, for large multibody systems, Newmark dissipative was the best election since, provided that the adequate parameters were chosen, excellent behavior was achieved in terms of efficiency, robustness and accuracy.

This paper represents a new step in the commented research. It is devoted to the study of the behavior of the already mentioned three dynamic formulations, when combined with the Runge-Kutta (RK) family of integrators, as an alternative to the structural integrators for real-time applications in multibody dynamics.

Two main types of RK integrators can be distinguished: explicit and implicit. The explicit RK integrators are simple and easy-to-use, but they cannot deal with stiff systems, very common in multibody dynamics due to both the presence of physical devices of high stiffness, and the consideration of the constraints by means of penalty techniques. The implicit RK integrators (IRK) are much more complex than their explicit counterparts, but they behave much better too.

Inside this group are the so-called Singly Diagonally Implicit Runge-Kutta (SDIRK) algorithms (see [Hairer and Wanner 1996], [Lambert 1997], [Ascher and Petzold 1998]). They show advantages with respect to the general IRK integrators regarding simplicity and computational cost and, moreover, they can provide good stability and accuracy properties. Therefore, the SDIRK integrators seem to be the most suited, belonging to the IRK family, to address real-time applications in multibody systems. This kind of applications will ask for a low number of stages, so that the computational cost of each time-step keeps moderate, in order to be competitive with the Newmark integrators, specifically adapted to the second order dynamic equations which arise in multibody systems.

It must be pointed out that two (global and hybrid) of the three dynamic formulations considered, state the equations of motion in dependent coordinates, thus leading to the integration of a system of differential algebraic equations (DAE). Therefore, for these two formulations, the approach is different from that followed in recent works also dealing with the use of RK integrators in multibody dynamics, like those of [Meijaard 2003] and [Negrut et al 2003], which derive the equations of motion in state-space form and, then, require the integration of a system of ordinary differential equations (ODE). Hence, the development of the equations that arise when combining a dynamic formulation in dependent coordinates and a SDIRK integrator, can be also considered as a contribution of this work.

The remaining of the paper is organized as follows. Section 2 offers a brief description of the three real-time dynamic formulations being compared, and highlights the changes that must be carried out in the two (global and hybrid) which state the equations of motion in dependent coordinates, in order to be used with IRK integrators. Section 3 proposes an adaptation of the general form of SDIRK integrators, aimed to make them more suitable for real-time purposes. In Section 4, the equations that result from the combination of the dynamic formulations considered and the SDIRK integrators are developed. Section 5 shows the algorithm followed in the implementation of the obtained methods. Section 6 addresses the selection of a specific SDIRK integrator, giving reasons to justify the authors' choice. Section 7 presents the example which served as benchmark, explains the procedure adopted to carry out the comparison among the three methods derived in the paper, as well as with those obtained in previous works, based on structural integrators, and provides the results of the study, which are discussed in Section 8. Finally, the conclusions of the work are summarized in Section 9.

2. THE PROPOSED DYNAMIC FORMULATIONS

The three formulations being compared have previously shown an excellent aptitude for real-time purposes [Cuadrado et al. 2004a], since they are efficient, i.e. they provide very fast calculation of the function evaluation, and they are robust, which means that they can perform large time-steps without losing stability. The first two will be called global and topological, respectively, and the third one will be called hybrid, since it was developed as a combination of the other two. A brief description of each formulation, as they appear in the two abovementioned previous works, is given in the next paragraphs, but further details can be found in the corresponding references.

The global method [Cuadrado et al. 2001] uses natural (global and dependent) coordinates to model the multibody system. It consists of an index-3 augmented Lagrangian formulation, whose equations are combined with the difference equations of the numerical integrator, to produce a non-linear algebraic set of equations wherein the dependent positions are the unknowns. Such set is solved through the Newton-Raphson iteration. Once convergence is attained at the time-step at position level, the corresponding velocities and accelerations, obtained through the difference equations of the numerical integrator, do not satisfy the first and second derivatives of the constraint equations, respectively. Therefore, in order to assure such fulfillment, they are cleaned by means of mass-damping-stiffness-orthogonal projections. The leading matrix of both projections, either in velocities as well as in accelerations, is the same, and coincident with the approximated tangent matrix used for the Newton-Raphson iteration, so eliminating the need of matrix triangularization to carry out the

projections. Due to its global character, the method is very easy-to-implement, but it is robust and efficient as well, provided sparse matrix techniques are applied.

The topological method [Garcia de Jalon et al. 2003], semi-recursive, begins with the opening of the closed-loops, so as to obtain the open-loop version of the mechanism. Then, a double set of coordinates is defined: six coordinates (three translations plus three rotations) for each body, and the relative coordinates of the whole open-loop version of the mechanism. The dynamic equations are expressed in the coordinates of the bodies and, then, a velocity transformation is carried out which leads to a set of motion equations in the relative coordinates, dependent in the general case. In order to calculate the leading matrix and the right-hand-side of that set of equations, an efficient recursive technique which accumulates forces and inertias is used. The following step consists of imposing the loop-closure conditions, in order to arrive at the state-space form of the equations of motion, which is achieved by a second velocity transformation, carried out by conventional non-recursive procedures. The state-space set of equations is combined with the difference equations of the numerical integrator, thus leading to a non-linear algebraic set of equations wherein the independent positions are the unknowns. The set is solved through the Newton-Raphson iteration, which provides the solution at position level for the current time-step. Since the variables are independent, there are no constraints to fulfill, and therefore the velocities and accelerations, obtained from the positions through the difference equations of the numerical integrator, are directly valid. Despite its topological character, the resulting method is reasonably easy-to-implement. Of course, it is also very efficient. Regarding the robustness, it suffers from the usual drawbacks of this technique: possible limitation in the range of validity of the independent set of coordinates selected, and lack of robustness in singular positions.

The hybrid method [Cuadrado et al. 2004a] shares the first stage with the topological method previously described, i.e. until the equations of motion are established in relative coordinates, dependent in general, for the open-loop version of the mechanism. However, once at such point, it turns to the procedure adopted for the global method: the index-3 augmented Lagrangian formulation is used to impose the loop-closure constraints. These equations of motion are combined with the difference equations of the numerical integrator, leading to a non-linear set of algebraic equations, wherein the dependent relative positions are the unknowns, which are obtained through the Newton-Raphson iteration. Then, mass-damping-stiffness orthogonal projections are performed at velocity and acceleration level so as to impose the first and second derivatives of the constraint equations. As in the global method, these two projections share the same leading matrix, which is coincident with the approximate tangent matrix used for the Newton-Raphson iteration, so that only forward reductions and back substitutions are needed to work out the projections. The easiness of implementation of this method is comparable to that of the topological method, while the robustness is improved, since the two drawbacks of the latter already mentioned in the previous paragraph are not longer present. Moreover, the hybrid method shows to be more efficient than both the global and topological methods for large multibody systems.

To summarize, remark that the global and hybrid formulations state the equations of motion in dependent coordinates (natural coordinates in the case of the global formulation, and relative coordinates in the case of the hybrid formulation), while the topological formulation derives the equations of motion in independent coordinates (some subset of the relative coordinates). Therefore, each formulation represents a different philosophy (global: a lot of dependent coordinates; hybrid: a few dependent coordinates; topological: independent coordinates), and this is why their comparison becomes interesting.

What has been described in the previous paragraphs is applicable when the three formulations are combined with the trapezoidal rule or, more generally, with structural integrators, since they enable to express the velocities and accelerations as functions of the positions and, therefore, it is possible to take the positions as primary variables, thus obtaining a nonlinear system of algebraic equations in which the positions are the unknowns. However, IRK integrators do not allow to express the velocities and accelerations as functions of the positions. This fact has an immediate consequence on the two formulations (global and hybrid) which state the equations of motion in dependent coordinates, since they are based on an augmented Lagrangian approach in index-3 form. In such approach, fulfillment of both the dynamic equations and the position constraints is achieved at the same time, during the iterative convergence process inside a time-step, but this requires that the integrator enables to express the velocities and accelerations as functions of the positions, which is not an option with the IRK integrators. Therefore, combination of the global and hybrid dynamic formulations with IRK integrators will demand the implementation of the augmented Lagrangian approach in its index-1 form. This means that satisfaction of both the dynamic equations and the acceleration (instead of position) constraints will be achieved at the same time during the iterative convergence process inside a time-step. Hence, in the case of the global and hybrid formulations, a change in their form is imposed by the use of the IRK integrators instead of the trapezoidal rule or, more generally, the structural integrators.

In what follows, the equations of motion for both the global and hybrid methods, i.e. for the two methods in dependent coordinates, are written. The difference is that, in the case of the global method, the vector of variables $\mathbf{q}$ represents the set of dependent natural coordinates, while in the case of the hybrid method the same vector represents the set of dependent relative coordinates. The case of the topological formulation is trivial,

since it uses independent coordinates and, therefore, doesn't include any constraint, which means that the equations of motion have the simple form $\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}$, where $\mathbf{q}$ represents the set of independent coordinates.

The equations of the motion provided by the mentioned index-1 augmented Lagrangian formulation are,

$$\left(\mathbf{M} + \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\alpha\mathbf{\Phi}_\mathbf{q}\right)\ddot{\mathbf{q}} = \mathbf{Q} - \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\alpha\left(\dot{\mathbf{\Phi}}_\mathbf{q}\dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_t\right) - \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\lambda}^* \tag{1}$$

where $\mathbf{q}$ is the vector of dependent coordinates of the mechanism, $\mathbf{M}$ is the mass matrix, $\mathbf{Q}$ is the vector of applied forces, $\mathbf{\Phi}$ is the vector of constraints, $\mathbf{\Phi}_\mathbf{q}$ is its Jacobian matrix, $\boldsymbol{\lambda}^*$ is the vector of Lagrange multipliers, and α is the penalty factor. In the case of the hybrid formulation, $\mathbf{M}$ and $\mathbf{Q}$ are obtained through an efficient recursive procedure, based on a velocity transformation technique. The vector of Lagrange multipliers is iteratively updated inside the time-step (sub-index *i*) according to the following expression,

$$\boldsymbol{\lambda}_{i+1}^* = \boldsymbol{\lambda}_i^* + \alpha\dot{\mathbf{\Phi}}_{i+1} \tag{2}$$

Once convergence is attained at the time-step, the resulting accelerations satisfy the second derivatives of the constraints, but the constraints themselves and their first derivatives are not satisfied by positions and velocities, respectively. To enforce such a fulfillment, projections of the positions and velocities are carried out. The form of the projections is, for the positions (iterative, sub-index *j*),

$$\left(\mathbf{M} + \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\alpha\mathbf{\Phi}_\mathbf{q}\right)_j \Delta\mathbf{q}_{j+1} = -\left(\mathbf{M}\Delta\mathbf{q}\right)_j - \left(\mathbf{\Phi}_\mathbf{q}^\mathrm{T}\alpha\mathbf{\Phi}\right)_j - \left(\mathbf{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\sigma}\right)_j \tag{3}$$

$$\boldsymbol{\sigma}_{j+1} = \boldsymbol{\sigma}_j + \alpha\mathbf{\Phi}_{j+1}$$

and, for the velocities (non-iterative),

$$\left(\mathbf{M} + \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\alpha}\mathbf{\Phi}_\mathbf{q}\right)\dot{\mathbf{q}} = \mathbf{M}\dot{\mathbf{q}}^* - \mathbf{\Phi}_\mathbf{q}^\mathrm{T}\boldsymbol{\alpha}\mathbf{\Phi}_t \tag{4}$$

where $\Delta\mathbf{q} = \mathbf{q} - \mathbf{q}^*$, and $\mathbf{q}^*, \dot{\mathbf{q}}^*$ are, respectively, the positions and velocities obtained after convergence is achieved at the time-step, which, as commented before, do not satisfy the constraints and their first derivatives. However, the new positions and velocities, $\mathbf{q}$ and $\dot{\mathbf{q}}$, worked out from Eqs. (3,4), do satisfy the constraints at position and velocity level.

3. GENERAL FORM OF SDIRK INTEGRATORS

The general equations of an *s*-stage SDIRK algorithm are well-known, and can be found in [Hairer and Wanner 1996] for the integration of first order ODE having the form $\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y})$. For a certain time-step starting at time t_0 and ending at time t_f, such equations are,

$$\mathbf{y}_f = \mathbf{y}_0 + h\sum_{i=1}^{s} b_i\mathbf{f}\left(t_0 + c_i h, \mathbf{y}_0 + \mathbf{z}_i\right) \tag{5}$$

where $\mathbf{y_0}$ and $\mathbf{y}_f$ are the state variables at times t_0 and t_f, respectively, $h = t_f - t_0$ is the time-step size, b_i and c_i are coefficients of the method, and the $\mathbf{z}_i$ are obtained from the following nonlinear set of equations,

$$\hat{\mathbf{z}} = \begin{pmatrix}\mathbf{z}_1\\ \mathbf{z}_2\\ \vdots\\ \mathbf{z}_s\end{pmatrix} = h\begin{pmatrix}\gamma\mathbf{I}_n & \mathbf{0} & \cdots & \mathbf{0}\\ a_{21}\mathbf{I}_n & \gamma\mathbf{I}_n & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ a_{s1}\mathbf{I}_n & a_{s2}\mathbf{I}_n & \cdots & \gamma\mathbf{I}_n\end{pmatrix}\begin{pmatrix}\mathbf{f}\left(t_0 + c_1 h, \mathbf{y}_0 + \mathbf{z}_1\right)\\ \mathbf{f}\left(t_0 + c_2 h, \mathbf{y}_0 + \mathbf{z}_2\right)\\ \vdots\\ \mathbf{f}\left(t_0 + c_s h, \mathbf{y}_0 + \mathbf{z}_s\right)\end{pmatrix} = h\mathbf{A}\hat{\mathbf{f}} \tag{6}$$

where γ and the a_{ij} are coefficients of the method, *n* is the number of state variables, and $\mathbf{I}_n$ is the identity matrix of size *n*.

Then, making use of Eq. (6), Eq. (5) can be rewritten as,

$$\mathbf{y}_f = \mathbf{y}_0 + h\mathbf{b}^\mathrm{T}\hat{\mathbf{f}} = \mathbf{y}_0 + \mathbf{b}^\mathrm{T}\mathbf{A}^{-1}\hat{\mathbf{z}} \tag{7}$$

where,

$$\mathbf{b}^\mathrm{T} = \begin{pmatrix}b_1\mathbf{I}_n & b_2\mathbf{I}_n & \cdots & b_s\mathbf{I}_n\end{pmatrix} \tag{8}$$

Let's define the state variables at each stage as,

$$\mathbf{y}_i = \mathbf{y}_0 + \mathbf{z}_i \tag{9}$$

and, consequently,

$$\hat{\mathbf{y}}^\mathrm{T} = \begin{pmatrix}\mathbf{y}_1^\mathrm{T} & \mathbf{y}_2^\mathrm{T} & \cdots & \mathbf{y}_s^\mathrm{T}\end{pmatrix} \tag{10}$$

In order to solve the nonlinear set of equations (6), the iterative Newton-Raphson procedure is applied, the residual vector being,

$$\mathbf{r} = \hat{\mathbf{z}} - h\mathbf{A}\hat{\mathbf{f}} \tag{11}$$

and the corresponding tangent matrix,

$$\mathbf{J} = \left(\frac{\partial \hat{\mathbf{z}}}{\partial \hat{\mathbf{z}}}\right) - h\mathbf{A}\left(\frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{z}}}\right) = \mathbf{I}_{sxn} - h\mathbf{A}\left(\frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{y}}}\right)\left(\frac{\partial \hat{\mathbf{y}}}{\partial \hat{\mathbf{z}}}\right) = \mathbf{I}_{sxn} - h\mathbf{A}\left(\frac{\partial \hat{\mathbf{f}}}{\partial \hat{\mathbf{y}}}\right) \tag{12}$$

where $\mathbf{I}_{sxn}$ stands for the identity matrix of size *s*x*n*.

The linear set of equations provided by the Newton-Raphson procedure is,

$$\mathbf{J} \cdot \Delta\hat{\mathbf{z}} = -\mathbf{r} \tag{13}$$

If each derivative in $\partial\hat{\mathbf{f}} / \partial\hat{\mathbf{y}}$ is approximated by $(\partial\mathbf{f} / \partial\mathbf{y})_0$, that is, by the value of the derivative at time t_0, then, Eq. (13) can be written as,

$$\begin{pmatrix} \mathbf{I}_n - \gamma h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 & \mathbf{0} & \cdots & \mathbf{0} \\ -a_{21}h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 & \mathbf{I}_n - \gamma h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ -a_{s1}h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 & -a_{s2}h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 & \cdots & \mathbf{I}_n - \gamma h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 \end{pmatrix} \begin{pmatrix} \Delta\mathbf{z}_1 \\ \Delta\mathbf{z}_2 \\ \vdots \\ \Delta\mathbf{z}_s \end{pmatrix} = -\left(\hat{\mathbf{z}} - h\mathbf{A}\hat{\mathbf{f}}\right) \tag{14}$$

and, finally, the unknowns are updated $\hat{\mathbf{z}} \leftarrow \hat{\mathbf{z}} + \Delta\hat{\mathbf{z}}$, until a certain error tolerance is achieved.

From Eq. (6), it is clear that each vector $\mathbf{z}_i$ only depends on itself and on the other vectors $\mathbf{z}_j$ such that *j*<*i*. Therefore, the linear system of equations (14) can be solved by blocks, thus requiring the solution of *s* blocks of size *n*x*n*, instead of a unique system of size (*s*x*n*)x(*s*x*n*). The equations corresponding to block *i* are,

$$\left(\mathbf{I}_n - \gamma h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0\right)\Delta\mathbf{z}_i = -\left(\mathbf{z}_i - h\sum_{j=1}^{i} a_{ij}\mathbf{f}_j - h\sum_{j=1}^{i-1} a_{ij}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 \Delta\mathbf{z}_j\right) \tag{15}$$

with,

$$\mathbf{f}_i = \mathbf{f}\left(t_0 + c_i h, \mathbf{y}_0 + \mathbf{z}_i\right) = \mathbf{f}\left(t_0 + c_i h, \mathbf{y}_i\right) \tag{16}$$

4. COMBINATION OF EQUATIONS OF MOTION AND INTEGRATOR

In this Section, the SDIRK integrator shown in Section 3 is to be combined with the dynamic formulation presented in Section 2, representing both the global and hybrid formulations. The equations for the topological formulation can be easily obtained as a particular case of those presented below, when all the terms related to the constraints are eliminated.

To reduce the second order of Eq. (1) to first order, so that the SDIRK integrators can be applied, positions and velocities are considered as state variables $\mathbf{y}^{\mathrm{T}} = \left(\mathbf{q}^{\mathrm{T}}, \dot{\mathbf{q}}^{\mathrm{T}}\right) = \left(\mathbf{q}^{\mathrm{T}}, \mathbf{p}^{\mathrm{T}}\right)$ and, then, Eq. (1) can be rewritten as,

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}} \end{bmatrix} \cdot \begin{Bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{p} \\ \mathbf{Q} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\mathbf{p} + \dot{\boldsymbol{\Phi}}_t\right) - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^* \end{Bmatrix} \tag{17}$$

or, in a more compact form, since $\dot{\mathbf{y}}^{\mathrm{T}} = \left(\dot{\mathbf{q}}^{\mathrm{T}}, \dot{\mathbf{p}}^{\mathrm{T}}\right) = \mathbf{f}\left(t, \mathbf{y}\right)^{\mathrm{T}}$,

$$\bar{\mathbf{M}} \cdot \mathbf{f} = \bar{\mathbf{Q}} \tag{18}$$

with,

$$\bar{\mathbf{M}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}} \end{bmatrix} \tag{19}$$

$$\bar{\mathbf{Q}} = \begin{Bmatrix} \mathbf{p} \\ \mathbf{Q} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\left(\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\mathbf{p} + \dot{\boldsymbol{\Phi}}_t\right) - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda}^* \end{Bmatrix} \tag{20}$$

Differentiating Eq. (18) and neglecting non-relevant terms, the following relation is obtained,

$$\bar{\mathbf{M}} \cdot \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{C} - \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}} \end{bmatrix} \tag{21}$$

Eq. (15) can be modified, so that the product $\bar{\mathbf{M}}\left(\partial\mathbf{f} / \partial\mathbf{y}\right)_0$ explicitly appears in it, thus avoiding the need of inverting the $\bar{\mathbf{M}}$ matrix to obtain $\left(\partial\mathbf{f} / \partial\mathbf{y}\right)$ in Eq. (21). Multiplication of Eq. (15) by $\bar{\mathbf{M}}$ yields,

$$\left(\bar{\mathbf{M}} - \gamma h\bar{\mathbf{M}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0\right)\Delta\mathbf{z}_i = -\left(\bar{\mathbf{M}}\mathbf{z}_i - h\sum_{j=1}^{i} a_{ij}\bar{\mathbf{M}}\mathbf{f}_j - h\sum_{j=1}^{i-1} a_{ij}\bar{\mathbf{M}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 \Delta\mathbf{z}_j\right) \tag{22}$$

If now the results of Eqs. (19,21) are substituted into Eq. (22), the following set of equations is obtained,

$$\begin{bmatrix} \mathbf{I} & -\gamma h\mathbf{I} \\ \gamma h\mathbf{K} & \mathbf{M}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}+\gamma h\left(\mathbf{C}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right) \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{z}_i^{\mathbf{q}} \\ \Delta\mathbf{z}_i^{\dot{\mathbf{q}}} \end{Bmatrix} =$$

$$-\begin{pmatrix} \mathbf{z}_i^{\mathbf{q}}-h\sum_{j=1}^{i}a_{ij}\dot{\mathbf{q}}_j-h\sum_{j=1}^{i-1}a_{ij}\Delta\mathbf{z}_j^{\dot{\mathbf{q}}} \\ \left(\mathbf{M}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}\right)\left(\mathbf{z}_i^{\dot{\mathbf{q}}}-h\sum_{j=1}^{i}a_{ij}\ddot{\mathbf{q}}_j\right)+h\sum_{j=1}^{i-1}a_{ij}\left(\mathbf{K}\Delta\mathbf{z}_j^{\mathbf{q}}+\left(\mathbf{C}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)\Delta\mathbf{z}_j^{\dot{\mathbf{q}}}\right) \end{pmatrix} = \begin{pmatrix} \mathbf{e}_i^{\mathbf{q}} \\ \mathbf{e}_i^{\dot{\mathbf{q}}} \end{pmatrix} \tag{23}$$

where the super-indices $\mathbf{q}$ and $\dot{\mathbf{q}}$ indicate that the corresponding variables are associated to positions and velocities, respectively.

Note that the mass, stiffness and damping matrices, as well as the vectors and matrices related to the constraints that appear in Eq. (23), must be evaluated at the beginning of the time-step, i.e. at time t_0.

The set of equations (23) can be decoupled if the first matrix equation is multiplied by the factor $(-\gamma h\mathbf{K})$, and then it is added to the second matrix equation, as proposed in [Meijaard 2003],

$$\left[\mathbf{M}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\boldsymbol{\Phi}_{\mathbf{q}}+\gamma h\left(\mathbf{C}+\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}}\right)+\left(\gamma h\right)^2\mathbf{K}\right]\Delta\mathbf{z}_i^{\dot{\mathbf{q}}}=\mathbf{e}_i^{\dot{\mathbf{q}}}-\gamma h\mathbf{K}\mathbf{e}_i^{\mathbf{q}} \tag{24}$$

$$\Delta\mathbf{z}_i^{\mathbf{q}}=\mathbf{e}_i^{\mathbf{q}}+\gamma h\Delta\mathbf{z}_i^{\dot{\mathbf{q}}} \tag{25}$$

Therefore, $\Delta\mathbf{z}_i^{\dot{\mathbf{q}}}$ can be obtained from Eq. (24), and then, its value can be introduced in the right-hand-side of Eq. (25), so as to get $\Delta\mathbf{z}_i^{\mathbf{q}}$.

5. IMPLEMENTATION

The method obtained in the previous Section has been implemented according to the following algorithm (once more, for the case of the topological formulation, make zero all the terms related to the constraints):

1- Start: t=0; $\boldsymbol{\lambda}^*=0$; $\mathbf{z}_i=0$, i=1,2,...,s; $\mathbf{q}$ and $\dot{\mathbf{q}}$ are known;
2- Solution of Eq. (1): $\ddot{\mathbf{q}}$;
3- Prediction of $\mathbf{z}_i$ i=1,2,...,s by means of Eq. (6);
4- Loop of times: t=t+h;
5- Calculation of the leading matrix of Eq. (24);
6- n=0;
7- Loop of Newton-Raphson iterations: n=n+1;
8- i=0;
9- Loop of stages: i=i+1;
10- $\mathbf{y}_i=\mathbf{y}_0+\mathbf{z}_i$; solution of Eq. (1): $\ddot{\mathbf{q}}_i$;
11- If $n>1$ then $\left(\boldsymbol{\lambda}_i^*\right)_n \leftarrow \left(\boldsymbol{\lambda}_i^*\right)_{n-1}+\alpha\left(\ddot{\boldsymbol{\Phi}}_i\right)_n$;
12- Calculation of $\mathbf{e}_i^{\mathbf{q}}, \mathbf{e}_i^{\dot{\mathbf{q}}}$ (see Eq. (23));
13- Solution of Eqs. (24) and (25): $\Delta\mathbf{z}_i$;
14- $\mathbf{z}_i \leftarrow \mathbf{z}_i+\Delta\mathbf{z}_i$;
15- If $i<s$ go to 9;
16- $\Delta\mathbf{y}=\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\Delta\hat{\mathbf{z}}$; $error=\left\|\Delta\mathbf{q}\right\|=\sqrt{\sum_{j=1}^{n/2}\Delta y_j^2}$;
17- If $error>tolerance$ go to 7;
18- Update the state variables according to Eq. (7): $\mathbf{y}_t \leftarrow \mathbf{y}_{t-h}+\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\hat{\mathbf{z}}$;
19- Projections of positions and velocities: Eqs. (3-4);
20- If $t<t_{\mathrm{end}}$ go to 4;
21- End.

Therefore, the main calculation effort of the algorithm can be summarized as follows: the leading matrix of Eq. (24) must be calculated at the beginning of each new time-step (step 5); the linear system of equations (1) must be solved once for each stage and iteration (step 10), so that the right-hand-side vector of Eq. (23) can be determined (step 12); the linear system of equations (24) has to be solved and then Eq. (25) has to be evaluated,

both once for each stage and iteration (step 13), but notice that the leading matrix of Eq. (24) is factorized only once at the beginning of the time-step (step 5).

6. SELECTION OF THE SDIRK INTEGRATOR

As commented in Section 1, when seeking for real-time performance, integrators must be selected which encompass a moderate computational cost along with good stability properties. Therefore, it seems that two-stage integrators will be the most suitable for real-time purposes among the SDIRK family, provided they exhibit a stable behavior when dealing with multibody systems.

The conditions that must be fulfilled by the coefficients of an IRK integrator for it to possess the first orders of accuracy, the so-called order conditions, are the following:

$$\sum_i b_i = 1 \qquad (1^{\text{st}} \text{ order}) \tag{26}$$

$$2\sum_{j,k} b_j a_{jk} = 1 \qquad (2^{\text{nd}} \text{ order}) \tag{27}$$

$$3\sum_{j,k,l} b_j a_{jk} a_{jl} = 1; \quad 6\sum_{j,k,l} b_j a_{jk} a_{kl} = 1 \qquad (3^{\text{rd}} \text{ order}) \tag{28}$$

Table 1 shows the values of the coefficients for two-stage SDIRK integrators, so that the order conditions are fulfilled up to the second order. The maximum order achievable by these two-stage methods is third order.

Table 1. *Second-order two-stage SDIRK integrators*.

$c_1=\gamma$	$a_{11}=\gamma$	$a_{12}=0$
$c_2=1-\gamma$	$a_{21}=1-2\gamma$	$a_{22}=\gamma$
	$b_1=1/2$	$b_2=1/2$

If the third order conditions given in (28) are imposed to the coefficients shown in Table 1, it comes out that the two conditions are reduced to the following single one:

$$b_2 a_{21}^2 = \frac{1}{3} - \gamma + \gamma^2 \tag{29}$$

Substituting now the values of b_2 and a_{21} provided by Table 1 in Eq. (29) yields,

$$\gamma^2 - \gamma + \frac{1}{6} = 0 \rightarrow \gamma = \frac{3 \pm \sqrt{3}}{6} \tag{30}$$

Then, there exist a couple of two-stage SDIRK methods which are third order accurate. The method with $\gamma = \left(3+\sqrt{3}\right)/6$ is A-stable, while the method with $\gamma = \left(3-\sqrt{3}\right)/6$ offers a very small stability area in the negative complex half-plane. Since the A-stability is a highly desirable property to deal with stiff systems, like those appearing in multibody systems, the first method can be a good candidate to be tested for real-time applications. However, it must be remarked that A-stability does not guarantees stability in the case of nonlinear systems, which is the case of multibody systems.

7. NUMERICAL EXAMPLE

As a good complex and realistic example to test methods for demanding real-time multibody applications, the full model of the military 4x4 Bombardier Iltis vehicle [Iltis 1990], illustrated in Fig. 1 and used as a benchmark problem by the European automobile industry to check multibody dynamic codes, has been chosen.

The vehicle has a total mass of about 1500 kg, and features four identical suspensions whose characteristics are the following:

- A nonlinear spring, which provides a force given by the nonlinear expression:
$$F_S = -4.0092 \cdot 10^6 + 2.8397 \cdot 10^7 x - 6.7061 \cdot 10^7 x^2 + 5.2796 \cdot 10^7 x^3 \ [N] \quad \text{for } x \text{ in } m$$
- A nonlinear damper, which provides a damping force given by the nonlinear expression:

$$\begin{aligned} F_D &= 9945.627v + 33955.72v^2 - 59832.25v^3 - 305651.0v^4 \ [N] && \text{for} \quad -0.2 < v < 0.21 \ m/s \\ F_D &= -416.42 + 1844.3v \ [N] && \text{for} \quad v < -0.2 \ m/s \\ F_D &= 1919.1638 + 1634.727v \ [N] && \text{for} \quad v > 0.21 \ m/s \end{aligned} \tag{31}$$

- A leaf spring, modeled as a linear spring of $35900 \ N/m$.
- A tire, whose radial stiffness is $460000 \ N/m$.

The simulation which has served to compare the different methods consists of 8 s of motion with the vehicle going up an inclined ramp and then down a series of stairs, starting at a horizontal speed of 5 m/s (the road

profile is shown in Fig. 2). A rather violent motion is undergone by the vehicle, reaching acceleration peaks of up to 5*g*.

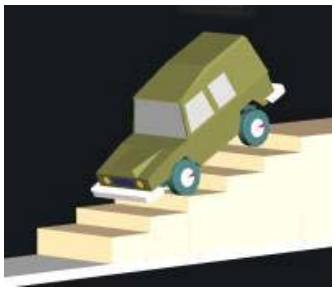

Figure 1. *The Iltis vehicle.*

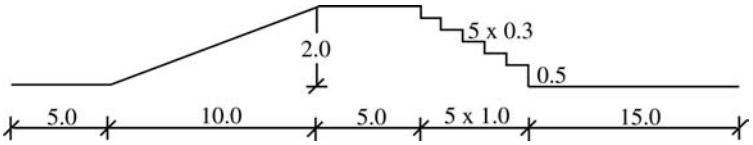

Figure 2. *Road profile.*

In order to establish the comparison among the three methods obtained from the combination of the global, topological and hybrid dynamic formulations, respectively, with the third order accurate, A-stable, two-stage SDIRK integrator ($\gamma = (3+\sqrt{3})/6$), programs to simulate the dynamics of the vehicle during the described maneuver according to each method have been implemented in FORTRAN language. Furthermore, the same simulation has been implemented in FORTRAN language through the methods obtained as combination of the three dynamic formulations with the trapezoidal rule. In this way, a comparison has been established between the performance offered by the global, topological and hybrid formulations when combined with either the trapezoidal rule or the SDIRK integrator. Since, in a previous work [Cuadrado et al. 2004b], the behavior of the three formulations with both the trapezoidal rule and the more general structural integrators (Newmark dissipative, HHT, Generalized-α) has been already compared, we will have the elements to judge whether the IRK integrators represent an alternative to the structural integrators for real-time applications in multibody dynamics. All the programs have been run on a PC with one AMD Athlon XP processor 1600+ @ 1.4 GHz.

Table 2. *Simulation results*.

	Time-step (s)	CPU-time (s)	# of iterations	$\frac{\text{Real-time}}{\text{CPU-time}}$
GLB+SDIRK	10^{-3}	---	---	0.001
	10^{-2}	---	---	
	10^{-4}	6705.13	427199	
GLB+TR	10^{-3}	133.48	57832	2.094
	10^{-2}	5.39	1909	
	$1.75 \cdot 10^{-2}$	3.82	1462	
TPL+SDIRK	10^{-3}	15.54	15492	1.822
	10^{-2}	4.39	4748	
	10^{-2}	4.39	4748	
TPL+TR	10^{-3}	4.11	8796	11.111
	10^{-2}	0.92	1876	
	$2.5 \cdot 10^{-2}$	0.72	1468	
HYB+SDIRK	10^{-3}	16.35	11538	1.848
	10^{-2}	---	---	
	$7.5 \cdot 10^{-3}$	4.33	3835	
HYB+TR	10^{-3}	5.75	9238	14.035
	10^{-2}	1.02	1839	
	$3.5 \cdot 10^{-2}$	0.57	1003	

8. DISCUSSION

The results provided by Table 2 are discussed in what follows.

First, it can be seen that, the cost per iteration is lower for TR methods than for SDIRK methods. This means that the function evaluation is performed faster in the first case. The reasons can be found if we pay attention to

the computational task undergone by each group of methods:

a) SDIRK methods must solve four linear systems of equations at each iteration, although two of them make use of a leading matrix already factorized at the beginning of the time-step (see Section 5); however, TR methods just need to solve one linear system of equations at each iteration.

b) While the linear system that must be solved at each iteration when using the TR approach is symmetric, two of the four linear systems that must be solved when using the SDIRK approach (see Eq. (23)), are not symmetric, due to the term $(\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\alpha\dot{\boldsymbol{\Phi}}_{\mathbf{q}})$.

c) The projections of the TR method take advantage of a leading matrix already factorized during the last Newton-Raphson iteration, while this is not the case for the SDIRK methods.

Reasons (b) and (c) are only applicable to those formulations including constraints, i.e. the global and hybrid formulations, which justifies the lower cost per iteration of the topological method with respect to the two others when combined with the SDIRK integrator.

Note that what has been explained occurs for the two-stage SDIRK integrator selected. For an SDIRK integrator having more stages, the disadvantage in computational effort per iteration with respect to the TR methods would increase.

Second, the TR methods show to be more stable than their SDIRK counterparts, as demonstrated by the larger time-steps they can reach. Among the SDIRK methods, it can be seen that the topological formulation provides the best stability properties, closely followed by the hybrid formulation, while the global formulation shows to be very limited. Hence, it seems that the methods present worse stability properties as the number of constraints increases (no constraints for the topological method, few constraints for the hybrid method, a lot of constraints for the global method). On the other hand, it must remarked that, when combined with the SDIRK integrator, the global and hybrid formulations are forced to state the equations of motion in their index-1 form, instead of the index-3 form used when combined with the trapezoidal rule (or other structural integrators). Undoubtedly, this is another factor which contributes to the reduced stability of such methods since, as shown in [Cuadrado et al. 1997], the index-3 scheme is largely more stable than the index-1 version.

It has been observed that, for the two formulations having constraints (global and hybrid), almost the same results are obtained with the SDIRK methods if the projections in positions and velocities at the end of each time-step are suppressed, while the projections in velocities and accelerations are essential for preserving the stability of the corresponding TR methods. However, the computational cost saved when eliminating the projections for the SDIRK methods, does not compensate at all its previously related drawbacks.

To summarize, the TR methods have shown to be largely more efficient and robust than their SDIRK counterparts (the difference in speed can be estimated in one order of magnitude). Taking into account that, as demonstrated in [Cuadrado et al. 2004b], TR methods are notably improved if the trapezoidal rule is substituted by, for example, an integrator from the Newmark dissipative family, it can be concluded that the IRK integrators are not competitive at all with the structural integrators to address the real-time dynamics of multibody systems.

9. CONCLUSIONS

Based on the obtained results, the following conclusions can be drawn from the present work:

- A study has been carried out to evaluate the aptitude of the IRK integrators for the real-time dynamics of multibody systems, and to compare it with that offered by the Newmark-type integrators.
- A third order, A-stable, two-stage SDIRK integrator has been considered as the most suitable from the IRK family for real-time purposes, due to the low computational effort it requires.
- Three dynamic formulations have been adapted in order to be combined with the mentioned SDIRK integrator: a global formulation in dependent natural coordinates, a topological formulation in independent relative coordinates, and a hybrid formulation in dependent relative coordinates. In the cases of the two formulations in dependent coordinates (global and hybrid), the original index-3 approach had to be turned into an index-1 scheme.
- The methods resulting from the combination of the three mentioned dynamic formulations and the SDIRK integrator selected have been developed, seeking always for the maximum efficiency, along with the way in which they must be implemented.
- A very demanding maneuver of a large, complex and realistic multibody system, the full model of the Iltis vehicle, has been programmed through the three dynamic formulations in combination with both the SDIRK integrator and the trapezoidal rule (TR), in order to establish a comparison among the different methods.
- The SDIRK methods have shown to be less efficient than the TR methods, due to their greater computational load at each time-step and, for the two formulations having constraints (global and hybrid), due also to the presence of a non-symmetric leading matrix in two of the four linear systems to be solved at each time-step.
- The SDIRK methods have shown to be less robust or stable than the TR methods. This effect is more acute as more constraints are considered (no constraints for the topological method, few constraints for the hybrid method, many constraints for the global method). For the two formulations in dependent coordinates (global and

hybrid), the use of the index-1 approach instead of the index-3 scheme adopted for the TR methods can also contribute to the reduced stability observed.

- The projections in positions and velocities at the end of each time-step can be suppressed for the SDIRK methods having constraints (global and hybrid), while the projections in velocities and accelerations are essential for the stability of the corresponding TR methods. However, this advantage in favor of the SDIRK methods does not compensate their previously described drawbacks.

- The advantage of the methods based on structural integrators would be larger if the SDIRK integrator was substituted by a different IRK integrator and/or the trapezoidal rule was substituted by a different structural integrator.

- Therefore, it comes out from this study that the IRK integrators are not competitive with the structural integrators to address the real-time dynamics of multibody systems.

ACKNOWLEDGMENT

This research has been sponsored by the Spanish MEC (Grant No. DPI2003-05547-C02-01) and the Galician DGID (Grant No. PGIDT04PXIC16601PN).

REFERENCES

- Ascher U.M. and Petzold L.R., *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Philadelphia Society for Industrial and Applied Mathematics, 1998.
- Cuadrado J., Cardenal J. and Bayo E., "Modeling and Solution Methods for Efficient Real-Time Simulation of Multibody Dynamics", *Multibody System Dynamics*, Vol. 1, No. 3, pp. 259-280, 1997.
- Cuadrado J., Gutierrez R., Naya M.A. and Morer P., "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*, Vol. 51, No. 9, pp. 1033-1052, 2001.
- Cuadrado J., Dopico D., Gonzalez M. and Naya M.A., "A Combined Penalty and Semi-Recursive Real-Time Formulation for Multibody Dynamics", *Journal of Mechanical Design*, to appear, 2004.
- Cuadrado J, Dopico D., Naya M.A. and Gonzalez M., "Penalty, Semi-Recursive and Hybrid Methods for MBS Real-Time Dynamics in the Context of Structural Integrators, *Multibody System Dynamics*, to appear, 2004.
- Garcia de Jalon J. and Bayo E., *Kinematic and Dynamic Simulation of Multibody Systems –The Real-Time Challenge–*, Springer-Verlag, 1994.
- Garcia de Jalon J., Alvarez E., de Ribera F.A., Rodriguez J.I. and Funes F.J., "Fast Dynamic Simulation of Multibody Systems based on a Double Velocity Transformation: Formulation and Implementation Details", *Multibody Dynamics*, Lisbon, Portugal, 2003.
- Geradin M. and Cardona A., *Flexible Multibody Dynamics. A Finite Element Approach*, Wiley, 2001.
- Hairer E. and Wanner G., *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer-Verlag, 1996.
- Iltis Data Package, *IAVSD Workshop*, Herbertov, Czechoslovakia, 1990.
- Lambert J.D., *Numerical Methods for Ordinary Differential Systems*, Wiley, 1997.
- Meijaard J.P., "Application of Runge-Kutta-Rosenbrock Methods to the Analysis of Flexible Multibody Systems", *Multibody System Dynamics*, Vol. 10, No. 3, pp. 263-288, 2003.
- Negrut D., Haug E.J. and German H.C., "An Implicit Runge-Kutta Method for Integration of Differential Algebraic Equations of Multibody Dynamics", *Multibody System Dynamics*, Vol. 9, No. 2, pp. 121-142, 2003.

BIOGRAPHIES

Daniel Dopico graduated in Mechanical Engineering from the University of La Coruña (Spain) in April 2002. Since May 2002 he works at the Laboratory of Mechanical Engineering where he enjoys a grant from the research funds of the Laboratory. He has been visiting scholar at the Universidad Politecnica de Madrid (Spain) from January to May 2004, and has got the doctor degree on October 2004. His work is focused on real-time formulations for multibody dynamics.

Urbano Lugris graduated in Mechanical Engineering from the University of La Coruña (Spain) in March 2003. Since May 2004 he is developing his doctorate at the Laboratory of Mechanical Engineering granted by the Spanish Ministry of Education and Science. His Ph.D. thesis addresses the topic of real-time formulations for flexible multibody dynamics.

Manuel Gonzalez got the degree as Mechanical Engineer in 1999 at the University of La Coruña. Since April 2000 he is Assistant Professor at the Department of Industrial Engineering II at the same university, and Ph. D. student at the Mechanical Engineering Laboratory in Ferrol. He has been visiting scholar at the Universidad Politecnica de Madrid (Spain) in 2001 and at the University of Illinois at Chicago (U.S.A.) in 2002. His research is focused on Computational Mechanics and CAE.

Javier Cuadrado graduated in Mechanical Engineering from the University of Navarra (Spain) in 1990. Afterwards, he worked as assistant professor at the School of Engineering of the same university and,

simultaneously, granted by the Spanish Ministry of Education and Science, he developed his doctorate at the department of Applied Mechanics of CEIT (Spain), a research center settled in the Basque Country, closely related to the University of Navarra. He received his Ph.D. degree in Mechanical Engineering from that university in 1993, having specialized in computational analysis of multibody systems. In 1994, he joined the Mechanical Engineering department of the University of La Coruña (Spain) as Associate Professor, becoming Full Professor in 2000. In March 2002, he founded the Laboratory of Mechanical Engineering at the mentioned department.

His research is oriented to the real-time computational kinematics and dynamics of multibody systems with rigid and flexible links. He has written more than 40 publications in the mentioned area. Currently, he is the thesis advisor of four doctoral students. He provides technical review for nine international technical journals. He is a member of the Asociacion Española de Ingenieria Mecanica (AEIM), the European Mechanics Society (EUROMECH), and the American Society of Mechanical Engineers (ASME).

He has collaborated in projects funded by the European Space Agency, US Army Research Office, and has participated in NATO sponsored Advanced Study Institutes. Currently, his research is supported by the Spanish and Galician Governments. He was member of the team that developed commercial software COMPAMM (COMPuter Analysis of Machines and Mechanisms), which was commercialized by Mechanical Dynamics Inc. all over the world.