IRK VS STRUCTURAL INTEGRATORS FOR REAL-TIME APPLICATIONS IN MBS

Daniel Dopico
Department of Mechanical Engineering
University of La Coruña
Ferrol, 15403, Spain
Email: ddopico@mail2.udc.es

Javier Cuadrado
Department of Mechanical Engineering
University of La Coruña
Ferrol, 15403, Spain
Email: javicuad@cdf.udc.es

ABSTRACT

Recently, the authors have developed a method for real-time dynamics of multibody systems, which combines a semi-recursive formulation to derive the equations of motion in dependent relative coordinates, along with an augmented Lagrangian technique to impose the loop closure conditions. The following numerical integration procedures, which can be grouped into the so-called structural integrators, were tested: trapezoidal rule, 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 described method 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

Newmark integrators, widely used in structural dynamics, have shown to adapt well to the equations of motion of multibody systems, even for real-time applications, as can be seen in Garcia de Jalon and Bayo (1994), or Geradin and Cardona (2001).

As an alternative, the suitability of the Runge-Kutta (RK) algorithms to face such kind of problems is being studied. 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.

2. THE PROPOSED DYNAMIC FORMULATION

The proposed dynamic formulation has been presented by the authors in Cuadrado and Dopico (2003). It combines a recursive technique to derive the equations of motion in relative coordinates, dependent in the general case, and an augmented Lagrangian approach to impose the loop closure conditions.

When combined with the structural integrators (Cuadrado and Dopico (2003)), the mentioned augmented Lagrangian approach was formulated in its index-3 form, so that fulfillment of both the dynamic equations and the position constraints was achieved at the same time, during the iterative convergence process inside a time-step. Such a formulation requires that the integrator enables to express the velocities and accelerations as functions of the positions, which in fact is possible when dealing with Newmark integrators, but not if IRK schemes are used. Therefore, combination of the proposed dynamic formulation 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 constraints will be achieved at the same

time during the iterative convergence process inside a time-step. Hence, a change in the formulation is imposed by the use of the IRK integrators instead of the Newmark integrators.

It must be pointed out that the proposed dynamic formulation states the equations of motion in relative coordinates, which are dependent in the general case, i.e. when the system presents closed-loops in its topology, thus leading to the integration of a system of differential algebraic equations (DAE). Therefore, the approach is different from that taken in recent works, like those of Meijaard (2003) and Negrut et al (2003), which state the equations of motion in state-space form and, then, require the integration of a system of ordinary differential equations (ODE).

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 relative 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, $\mathbf{\Phi_q}$ is its Jacobian matrix, $\boldsymbol{\lambda}^*$ is the vector of Lagrange multipliers, and α is the penalty factor. $\mathbf{M}$ and $\mathbf{Q}$ are obtained through an efficient recursive procedure, based on a velocity transformation technique. The vector of Lagrange multipliers is iteratively updated (sub-index *i*) according to the following expression,

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

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*),

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

and, for the velocities (non-iterative),

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

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 above, do not satisfy the constraints and their first derivatives.

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_1, such equations are,

$$\mathbf{y_1}=\mathbf{y_0}+h\sum_{i=1}^{s}b_i\mathbf{f}\left(t_0+c_ih,\mathbf{y_0}+\mathbf{z}_i\right) \tag{5}$$

where $\mathbf{y_0}$ and $\mathbf{y_1}$ are the state variables at times t_0 and t_1, respectively, $h=t_1-t_0$ 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,

$$\hat{\mathbf{z}}=\begin{pmatrix}\mathbf{z_1}\\ \mathbf{z_2}\\ \vdots\\ \mathbf{z_s}\end{pmatrix}=h\begin{pmatrix}\gamma\mathbf{I_n} & \mathbf{0} & \cdots & \mathbf{0}\\ a_{21}\mathbf{I_n} & \gamma\mathbf{I_n} & \cdots & \mathbf{0}\\ \vdots & \vdots & & \vdots\\ a_{s1}\mathbf{I_n} & a_{s2}\mathbf{I_n} & \cdots & \gamma\mathbf{I_n}\end{pmatrix}\begin{pmatrix}\mathbf{f}\left(t_0+c_1h,\mathbf{y_0}+\mathbf{z_1}\right)\\ \mathbf{f}\left(t_0+c_2h,\mathbf{y_0}+\mathbf{z_2}\right)\\ \vdots\\ \mathbf{f}\left(t_0+c_sh,\mathbf{y_0}+\mathbf{z_s}\right)\end{pmatrix}=h\mathbf{A}\hat{\mathbf{f}}$$

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,

$$\mathbf{y_1}=\mathbf{y_0}+h\mathbf{b}^\mathbf{T}\hat{\mathbf{f}}=\mathbf{y_0}+\mathbf{b}^\mathbf{T}\mathbf{A}^{-1}\hat{\mathbf{z}} \tag{7}$$

where,

$$\mathbf{b}^\mathbf{T}=\begin{pmatrix}b_1\mathbf{1_{1xn}} & b_2\mathbf{1_{1xn}} & \cdots & b_s\mathbf{1_{1xn}}\end{pmatrix} \tag{8}$$

with $\mathbf{1_{1xn}}$ representing a row vector of ones.

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

$$\mathbf{r}=\hat{\mathbf{z}}-h\mathbf{A}\hat{\mathbf{f}} \tag{9}$$

and the corresponding approximated tangent matrix,

$$\mathbf{J}=\left(\frac{\partial\hat{\mathbf{z}}}{\partial\hat{\mathbf{z}}}\right)-h\mathbf{A}\left(\frac{\partial\hat{\mathbf{f}}}{\partial\hat{\mathbf{z}}}\right)=\mathbf{I_{sxn}}-h\mathbf{A}\left(\frac{\partial\mathbf{f}}{\partial\hat{\mathbf{y}}}\right)\left(\frac{\partial\mathbf{y}}{\partial\mathbf{z}}\right)=\mathbf{I_{sxn}}-h\mathbf{A}\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right) \tag{10}$$

where $\mathbf{I_{sxn}}$ stands for the identity matrix of size *sxn*.

If the derivatives $\partial\hat{\mathbf{f}}/\partial\hat{\mathbf{y}}$ are approximated by $\left(\partial\mathbf{f}/\partial\mathbf{y}\right)_\mathbf{0}$, that is, by the value of the derivative at time t_0, then, the linear set of equations provided by the Newton-Raphson procedure is,

$$\begin{pmatrix}\mathbf{I_n}-\gamma h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}\\ -a_{21}h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0} & \mathbf{I_n}-\gamma h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ -a_{s1}h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0} & -a_{s2}h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0} & \cdots & \mathbf{I_n}-\gamma h\left(\frac{\partial\mathbf{f}}{\partial\mathbf{y}}\right)_\mathbf{0}\end{pmatrix}\begin{pmatrix}\Delta\mathbf{z_1}\\ \Delta\mathbf{z_2}\\ \vdots\\ \Delta\mathbf{z_s}\end{pmatrix}=-\begin{pmatrix}\mathbf{z_1}-h\mathbf{A_1f_1}\\ \mathbf{z_2}-h\mathbf{A_2f_2}\\ \vdots\\ \mathbf{z_s}-h\mathbf{A_sf_s}\end{pmatrix}$$

with,

$$\mathbf{A_i}=\begin{bmatrix}a_{i1}\mathbf{I_n} & a_{i2}\mathbf{I_n} & \ldots & \gamma\mathbf{I_n} & \mathbf{0} & \ldots & \mathbf{0}\end{bmatrix} \tag{12}$$

$$\mathbf{f_i}=\mathbf{f}\left(t_0+c_ih,\mathbf{y_0}+\mathbf{z_i}\right) \tag{13}$$

and, finally, the unknowns are updated $\hat{\mathbf{z}}\leftarrow\mathbf{z}+\Delta\mathbf{z}$, until a

certain error tolerance is achieved.

From Eq. (11), 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 (11) can be solved by blocks, thus requiring the solution of s blocks of size nxn, instead of a unique system of size (sxn)x(sxn). The equations corresponding to block i are,

$$\left(\mathbf{I}_n - \gamma h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0\right)\Delta\mathbf{z}_i = $$

$$= -\left(\mathbf{z}_i - \sum_{j=1}^{i} h a_{ij}\mathbf{f}_j - \sum_{j=1}^{i-1} a_{ij} h\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 \Delta\mathbf{z}_j\right) \tag{14}$$

4. COMBINATION OF EQUATIONS OF MOTION AND INTEGRATOR

In this Section, the SDIRK integrators shown in Section 3 are to be combined with the dynamic formulation presented in Section 2.

In order 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 $\mathbf{y}^T = \left(\mathbf{q}^T, \dot{\mathbf{q}}^T\right) = \left(\mathbf{q}^T, \mathbf{p}^T\right)$ and, therefore, Eq. (1) can be rewritten as,

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} + \boldsymbol{\Phi}_\mathbf{q}^T \alpha \boldsymbol{\Phi}_\mathbf{q} \end{bmatrix} \cdot \begin{Bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{Bmatrix} =$$

$$= \begin{Bmatrix} \mathbf{p} \\ \mathbf{Q} - \boldsymbol{\Phi}_\mathbf{q}^T \alpha\left(\dot{\boldsymbol{\Phi}}_\mathbf{q}\mathbf{p} + \dot{\boldsymbol{\Phi}}_\mathbf{t}\right) - \boldsymbol{\Phi}_\mathbf{q}^T \boldsymbol{\lambda}^* \end{Bmatrix} \tag{15}$$

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

$$\bar{\mathbf{M}} \cdot \mathbf{f} = \bar{\mathbf{Q}} \tag{16}$$

with,

$$\bar{\mathbf{M}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} + \boldsymbol{\Phi}_\mathbf{q}^T \alpha \boldsymbol{\Phi}_\mathbf{q} \end{bmatrix} \tag{17}$$

Differentiating Eq. (16) and neglecting non-relevant terms, the following relation is obtained,

$$\bar{\mathbf{M}} \cdot \left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{C} - \boldsymbol{\Phi}_\mathbf{q}^T \alpha \dot{\boldsymbol{\Phi}}_\mathbf{q} \end{bmatrix} \tag{18}$$

Eq. (14) 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. (18). Multiplication of Eq. (14) by $\bar{\mathbf{M}}$ yields,

$$\left(\bar{\mathbf{M}} - \gamma h \bar{\mathbf{M}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0\right)\Delta\mathbf{z}_i = = -\left(\bar{\mathbf{M}}\mathbf{z}_i - \sum_{j=1}^{i} h a_{ij}\bar{\mathbf{M}}\mathbf{f}_j - \sum_{j=1}^{i-1} a_{ij} h \bar{\mathbf{M}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right)_0 \Delta\mathbf{z}_j\right) \tag{19}$$

If now the results of Eqs. (17-18) are substituted into Eq. (19), the following set of equations is obtained,

$$\begin{bmatrix} \mathbf{I} & -\gamma h \mathbf{I} \\ \gamma h \mathbf{K} & \mathbf{M} + \boldsymbol{\Phi}_\mathbf{q}^T \alpha \boldsymbol{\Phi}_\mathbf{q} + \gamma h\left(\mathbf{C} + \boldsymbol{\Phi}_\mathbf{q}^T \alpha \dot{\boldsymbol{\Phi}}_\mathbf{q}\right) \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{z}_i^\mathbf{q} \\ \Delta\mathbf{z}_i^{\dot{\mathbf{q}}} \end{Bmatrix} = = -\begin{pmatrix} \mathbf{z}_i^\mathbf{q} \\ \left(\mathbf{M} + \boldsymbol{\Phi}_\mathbf{q}^T \alpha \boldsymbol{\Phi}_\mathbf{q}\right)\left(\mathbf{z}_i^\dot{\mathbf{q}} - \sum_{j=1}^{i} \right. \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{e}_i^\mathbf{q} \\ \mathbf{e}_i^{\dot{\mathbf{q}}} \end{pmatrix} \tag{20}$$

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

The set of equations (20) 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),

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

$$\Delta\mathbf{z}_i^\mathbf{q} = \mathbf{e}_i^\mathbf{q} + \gamma h \Delta\mathbf{z}_i^{\dot{\mathbf{q}}} \tag{22}$$

Therefore, $\Delta\mathbf{z}_i^{\dot{\mathbf{q}}}$ can be obtained from Eq. (21), and then, its value can be introduced in the right-hand-side of Eq. (22), so as to get $\Delta\mathbf{z}_i^\mathbf{q}$.

5. IMPLEMENTATION

The proposed method obtained as combination of the described index-1 augmented Lagrangian formulation and the SDIRK family of integrators, has been implemented according to the following algorithm:

1- Start: t=0, $\boldsymbol{\lambda}^* = 0$, $\mathbf{q}$ and $\dot{\mathbf{q}}$ are known;

2- Solution of Eq. (1): $\ddot{\mathbf{q}}$;

3- Prediction of $\mathbf{z}_i$ i=1,s by means of Eq. (6);

4- Loop of times: t=t+h;

5- Calculation of the tangent matrix of Eq. (21);

6- n=0;

7- Loop of Newton-Raphson iterations: n=n+1;

8- i=0;

9- Loop of stages: i=i+1;

10- Solution of Eq. (1);

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. (20));

13- Solution of Eqs. (21) and (22): $\Delta\mathbf{z}_i$;

14- $\mathbf{z}_i \leftarrow \mathbf{z}_i + \Delta\mathbf{z}_i$;

15- If $i < s$ go to 9 ;

16- $error = \sum_{i=1}^{s} \left\| \Delta \mathbf{z}_i \right\|$

17- If $error > tolerance$ go to 7 ;

18- Update the state variables according to Eq. (7): $\mathbf{y}_t \leftarrow \mathbf{y}_{t-1} + \mathbf{b}^T \mathbf{A}^{-1} \hat{\mathbf{z}}$;

19- Projections of positions and velocities: Eqs. (3-4);

20- If $t < t_{end}$ go to 4 ;

21- End.

Therefore, the main calculation effort of the algorithm can be summarized as follows: the tangent matrix of Eq. (21) 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 residual vector of Eq. (20) can be determined (step 12); the linear system of equations (21) has to be solved and then Eq. (22) has to be evaluated, both once for each stage and iteration, (step 13), but notice that the leading matrix of Eq. (21) is factorized only once at the beginning of the time-step (step 5).

6. SELECTION OF THE SDIRK INTEGRATOR

As commented before, 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 fulfill the coefficients of an IRK integrator for it to possess the first orders of accuracy, the so-called order conditions, are the following:

$$\sum_i b_i = 1 \qquad (1^{\text{st}} \text{ order}) \tag{23}$$

$$2\sum_{j,k} b_j a_{jk} = 1 \qquad (2^{\text{nd}} \text{ order}) \tag{24}$$

$$3\sum_{j,k,l} b_j a_{jk} a_{jl} = 1; \quad 6\sum_{j,k,l} b_j a_{jk} a_{kl} = 1 \qquad (3^{\text{rd}} \text{ order}) \tag{25}$$

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. 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 (25) are imposed to the coefficients shown in Table 1, it comes out that the two conditions are reduced to the following single one:

$$b_2 a_{21}^2 = \frac{1}{3} - \gamma + \gamma^2 \tag{26}$$

Substituting now the values of b_2 and a_{21} provided by Table 1 in Eq. (26) yields,

$$\gamma^2 - \gamma + \frac{1}{6} = 0 \rightarrow \gamma = \frac{3 \pm \sqrt{3}}{6} \tag{27}$$

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

In order to test the proposed formulation for demanding real-time multibody applications, a large, complex and realistic example, the full model of the 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 analyzed.

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 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 $5g$.

Programs to simulate the dynamics of the vehicle through the two following methods have been implemented in FORTRAN language:

a) The approach proposed in this paper, that is, a semi-

recursive formulation in relative coordinates, dependent in the general case, which imposes the loop closure conditions through an index-1 augmented Lagrangian formulation with projections in positions and velocities, combined with a third order accurate, A-stable, two-stage SDIRK integrator, having a value of $\gamma = \left(3+\sqrt{3}\right)/6$. This approach will be referred to as SDIRK.

b) The approach proposed in Cuadrado and Dopico (2003), that is, the same semi-recursive formulation in relative coordinates mentioned in the previous paragraph, but this time imposing the loop closure conditions through an index-3 augmented Lagrangian formulation with projections in velocities and accelerations, and combined with the trapezoidal rule, which is the simplest structural integrator. This approach will be referred to as TR.

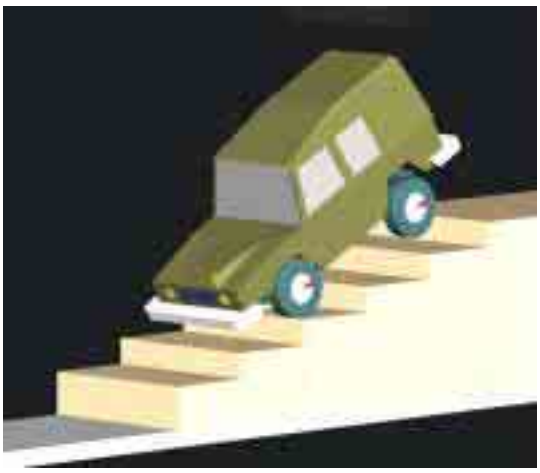

Figure 1. THE ILTIS VEHICLE.

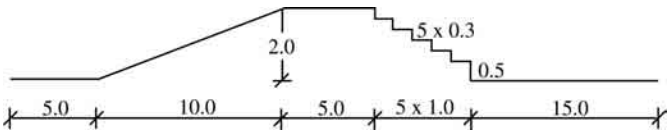

Figure 2. ROAD PROFILE.

The programs have been run on a PC with one AMD Athlon XP processor 1600+ @ 1.4 GHz.

8. RESULTS AND DISCUSSION

First, the two methods have been compared for a time-step size of 0.001 s. The TR method has shown to be 2.9 times faster than the SDIRK method, while providing a similar level of accuracy, although the number of required iterations is more or less the same for both methods: 9200 iterations for the TR method; 10899 for the SDIRK method.

The difference in efficiency can be easily explained if the calculation load of each method is roughly reminded: the SDIRK method must solve four linear systems of equations for each iteration, although two of them make use of a leading matrix already factorized at the beginning of the time-step (see Section 5); the TR method just need to solve one linear system of equations for each iteration.

Moreover, while the linear system that must be solved for each iteration when using the TR approach is symmetric, two out of the four linear systems that must be solved at each iteration when using the SDIRK approach (see Eq. (21)), are not symmetric, due to the term $(\mathbf{\Phi}_\mathbf{q}^\mathbf{T}\alpha\dot{\mathbf{\Phi}}_\mathbf{q})$. This fact also contributes to decrease the efficiency of the SDIRK method.

Therefore, it is clear that the advantage in efficiency of the TR method would be larger in case that the selected SDIRK integrator possessed more than two stages.

Second, the time-step size has been increased for each method, until the corresponding limit of convergence is reached, so as to evaluate the stability properties. The SDIRK method reached a maximum time-step size of 0.0075 s, providing an over-oscillating solution, which announced the imminent loss of stability. The TR method reached a maximum time-step size of 0.035 s, almost five times larger than that of the SDIRK method and, moreover, the solution did not show over-oscillation. The reason for such different behaviors can be the index-3 approach taken for the TR method, face to the index-1 approach followed for the SDIRK method. But remind that adopting the index-3 approach for the SDIRK method was not possible.

On the favor of the SDIRK method, it must be said that almost the same results are obtained 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 TR method. However, the computational cost saved when eliminating the projections for the SDIRK method, does not compensate at all its previously related drawbacks.

Consequently, the TR method has shown to be largely more efficient and robust than the SDIRK method. Furthermore, note that, as demonstrated in Cuadrado and Dopico (2003), the TR method was notably improved if the trapezoidal rule was substituted by, for example, an integrator from the Newmark dissipative family. Therefore, it can be concluded that the IRK integrators are not competitive at all with the structural integrators, when addressing 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 method for the real-time dynamics of multibody systems has been developed, as combination of a semi-recursive index-1 augmented Lagrangian formulation in relative coordinates, and a third order accurate, A-stable, two-stage SDIRK integrator. The method has been designed so as to keep at a minimum the computational cost required.
- The method has been compared with another one, obtained as combination of a semi-recursive index-3 augmented Lagrangian formulation in relative coordinates, and the simplest structural integrator, the trapezoidal rule.
- The comparison has been established through the simulation of a very demanding maneuver of a large, complex and realistic multibody system: the full model of the Iltis vehicle.
- The IRK method has shown to be less efficient than its structural counterpart, due to a greater computational load for each time-step, and to the presence of a non-symmetric leading matrix in two out of the four linear systems to be solved for each time-step.
- The IRK method has shown to be less robust or stable than its competitor, maybe due to the index-1 approach required in the former, face to the index-3 approach adopted in the latter.
- The projections in positions and velocities at the end of each time-step can be suppressed for the IRK method, while the corresponding projections in velocities and accelerations are essential for the stability of the structural method. However, this advantage in favor of the IRK method does not compensate its previously described drawbacks.
- The advantage of the structural method with respect to the IRK method would be larger if a different IRK integrator and/or a different structural integrator was used.
- 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 CICYT (Grant No. DPI2000-0379) and the Galician SGID (Grant No. PGIDT01PXI16601PN).

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. and Dopico D., Penalty, semi-recursive and hybrid methods for MBS real-time dynamics in the context of structural integrators, *Multibody Dynamics*, Proceedings CD, Lisbon, Portugal, 2003.

Garcia de Jalon J. and Bayo E., *Kinematic and dynamic simulation of multibody systems*, Springer-Verlag, 1994.

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-Rosenbroc k methods to the analysis of flexible multibody systems, *Multibody System Dynamics*, Vol. 10, 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, pp. 121-142, 2003.