Archive of Applied Mechanics 66 (1996) 187–199 © Springer-Verlag 1996

Symbolic generation of large multibody system dynamic equations using a new semi-explicit Newton/Euler recursive scheme

P. Fisette, J. C. Samin

Summary The aim of this paper is to show that multibody systems with a large number of degrees of freedom can be efficiently modelled, taking conjointly advantage of a recursive formulation of the equations of motion and of the symbolic generation capabilities.

Recursive schemes are widely used in the field of multibody dynamics since they avoid the "explosion" of the number of arithmetical operations in case of large multibody models. Within the context of our field of applications (railway dynamics simulation), explicit integration schemes are still prefered and thus oblige us to compute the generalized accelerations at each time step. To achieve this, we propose a new formulation of the well-known Newton/Euler recursive method, whose efficiency will be compared with a so-called "O(N)" formulation.

A regards the symbolic generation, often decried due to the size of the equations in case of large systems, we have recently implemented recursive multibody formalisms in the symbolic programme ROBOTRAN [1]. As we shall explain, the recursive nature of these formalisms is particularly well-suited to symbolic manipulation.

All these developments have been successfully applied in the field of railway dynamics, and in particular allowed us to analyse the dynamic behaviour of several railway vehicles. Some typical results related to a completely non-conventional bogie will be presented before concluding.

Key words dynamic modelling, multibody system, symbolic generation, computer simulation

1 Introduction

The multibody system dynamic analysis involves several modelling steps among which it is necessary to clearly distinguish the phase of *generation* of the equations of motion from the phase of *resolution* of the latter. Indeed, some confusion frequently appears concerning these two tasks, especially when evaluating a method in terms of computer efficiency (i.e. the number of arithmetic operations required for a given result, [2]).

As regards the *generation*, several theoretical formalisms are suitable to obtain the equations of motion in their scalar form. Moreover, the choice between absolute, relative or mixed coordinates must be considered with regard to the envisaged applications and could be discussed here. However, within the context of this paper, we only consider the relative coordinates approaches, and we distinguish those based on a virtual principle [3, 4], from those which directly use the classical Newton/Euler equations of motion in a recursive form [5, 6]. If the former have been successfully used in several applications such as spacecraft or mechanisms analysis, the latter seem to be particularly well-suited to large multibody systems, thanks to their recursive nature. We have finally adopted such formalisms in relative coordinates, in particular to deal with railway vehicle dynamics.

Another important aspect of the generation process is purely of a computer nature, and concerns, the way the system of equations is obtained: numerically or symbolically. The latter method exhibits several advantages over the former, the most significant being the reduction of mathematical expressions to be computed, and the absence of an algorithmic reconstruction of the equations of

Received 31 October 1994; accepted for publication 24 June 1995

P. Fisette[1], J.-C. Samin
Catholic University of Louvain-la-Neuve, Mechanical Department,
1348 Louvain-la-Neuve, Belgium

[1] Chargé de recherches du F.N.R.S.

motion at each step of a given numerical procedure. In the multibody dynamics domain, several symbolic programmes have been developed and cited in [1] at the same rank as numerical multibody softwares.

We have definitively adopted the symbolic approach in multibody simulation since our last developments in the ROBOTRAN software [7] allow us to generate without any difficulties the symbolic equations of motion of large multibody systems, on the basis of recursive formalisms.

Concerning the *resolution* of the equations of motion, and in particular their numerical integration, both the final form of the equations and the numerical method to integrate them have to be considered. In case of constrained multibody systems – which certainly represent the major part of the possible applications – the mathematical model consists of a differential/algebraic system ("DAE" system) which cannot be solved by classical integrators. A first possibility is to reduce the DAE system to a purely differential one ("ODE" system), compatible with classical schemes. An alternative consists in directly solving the global DAE system with an implicit integration scheme [8, 9].

At the present time, we have opted for a well-known reduction procedure, based on the "coordinate partitioning method" [10], which, at the expense of some CPU time penalty, is particularly reliable for the kinematic loop closure. This specific aspect of the modelling is fundamental for the geometrical problem of wheel/rail contact in railway dynamics [11], which represents the major part of our multibody applications.

2
Multibody formalism: a semi-explicit recursive scheme

The choice of a multibody formalism is governed by numerous aspects such as the field of applications, the type of coordinates, the computer implementation and the desired numerical analysis. In our case, the complex articulated structure of the railway bogies we had to analyse (see Fig. 1) induced us to keep a relative coordinates approach, in accordance with the ROBOTRAN programme philosophy [1], and to generate the equations of motion on the basis of the well-known "Recursive Newton/Euler Method". This latter has been initially developed in robotics for the inverse dynamic problem [12, 13]. Indeed, its recursive character allows to compute in a minimum of arithmetic operations the generalized forces Q to be applied to the joints, as a function of the generalized joint positions q, velocities $\dot{q}$ and accelerations $\ddot{q}$

$$Q = Q(q, \dot{q}, \ddot{q}). \tag{1}$$

Since the efficiency of the method is also attractive for simulation purposes, we have modified the original scheme which provides the equations of motion in their implicit form (1), in order to obtain recursively the following semi-explicit form for an unconstrained system:

$$M(q)\ddot{q} + C(q, \dot{q}) = Q(q, \dot{q}), \tag{2}$$

where: M is the symmetric generalized mass matrix, C contains the Coriolis, centrifugal and gravity terms as well as external forces and torques.

2.1
Forward kinematics

Consider in Fig. 2 a rigid body i carried by a rigid body h via a joint i, and let assume that every joint of the system has only one d.o.f. (degree of freedom): revolute or prismatic. This hypothesis, underlying

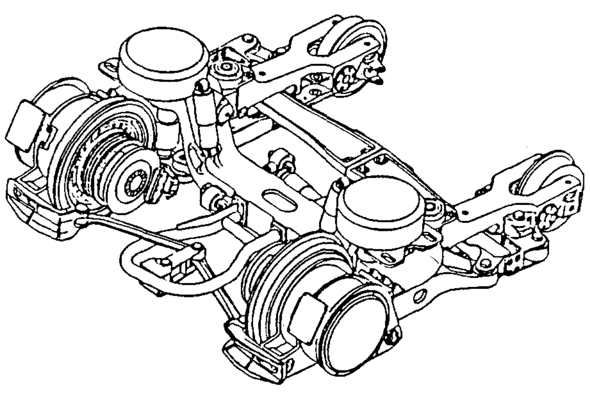


Fig. 1. BAS 2000 articulated bogie (B.N. Eurorail-Belgium)

the ROBOTRAN programme, allows without any restriction to model up to six d.o.f. joints by using intermediate fictitious massless bodies. For the i^{th} joint, let first define the unit vectors $\mathbf{\Phi}^i$ and $\mathbf{\Psi}^i$ such that

- for a prismatic joint, $\mathbf{\Psi}^i$ is the unit vector along the translational direction and $\mathbf{\Phi}^i = 0$,
- for a revolute joint, $\mathbf{\Phi}^i$ is the unit vector along the axis of rotation and $\mathbf{\Psi}^i = 0$.

We also define $\mathbf{z}^i \triangleq q^i\,\mathbf{\Psi}^i$, the relative displacement vector in the (prismatic) joint i, and $\mathbf{\Omega}^i \triangleq \dot{q}^i\,\mathbf{\Phi}^i$, the relative angular velocity vector associated with the (revolute) joint i.

For body i, one can write (see Fig. 2):

Absolute positions

$\mathbf{p}^i = \mathbf{p}^h + \mathbf{p}_z^{hi}$ for the i^{th} joint

$\mathbf{x}^i = \mathbf{p}^i + \mathbf{l}^{ii}$ for the centre of mass of body i.

Absolute velocities:

angular: $\boldsymbol{\omega}^i = \boldsymbol{\omega}^h + \mathbf{\Omega}^i = \boldsymbol{\omega}^h + \dot{q}^i\mathbf{\Phi}^i$ for body i, (3)

linear: $\dot{\mathbf{p}}^i = \dot{\mathbf{p}}^h + \tilde{\boldsymbol{\omega}}^h\cdot\mathbf{p}_z^{hi} + \dot{q}^i\mathbf{\Psi}^i$, (4)

$\dot{\mathbf{x}}^i = \dot{\mathbf{p}}^i + \tilde{\boldsymbol{\omega}}^i\cdot\mathbf{l}^{ii}$, (5)

Absolute accelerations:

angular: $\dot{\boldsymbol{\omega}}^i = \dot{\boldsymbol{\omega}}^h + \tilde{\boldsymbol{\omega}}^i\cdot\mathbf{\Phi}^i\dot{q}^i + \ddot{q}^i\mathbf{\Phi}^i$, (6)

linear: $\ddot{\mathbf{p}}^i = \ddot{\mathbf{p}}^h + (\tilde{\dot{\boldsymbol{\omega}}}^h + \tilde{\boldsymbol{\omega}}^h\tilde{\boldsymbol{\omega}}^h)\cdot\mathbf{p}_z^{hi} + 2\tilde{\boldsymbol{\omega}}^i\cdot\mathbf{\Psi}^i\dot{q}^i + \ddot{q}^i\mathbf{\Psi}^i$, (7)

$\ddot{\mathbf{x}}^i = \ddot{\mathbf{p}}^i + (\tilde{\dot{\boldsymbol{\omega}}}^i + \tilde{\boldsymbol{\omega}}^i\tilde{\boldsymbol{\omega}}^i)\cdot\mathbf{l}^{ii}$. (8)

In order to obtain the dynamic equations in a compact form, it is convenient to define the following quantities:

$\boldsymbol{\beta}^i \triangleq \tilde{\dot{\boldsymbol{\omega}}}^i + \tilde{\boldsymbol{\omega}}^i\tilde{\boldsymbol{\omega}}^i$, (9)

and

$\boldsymbol{\alpha}^i \triangleq \ddot{\mathbf{p}}^i - \mathbf{g}$. (10)

The forward kinematic recursion of the classical Newton/Euler recursive scheme can then be written in a vectorial form as follows:

Initialisation:

$\boldsymbol{\omega}^0 = 0, \quad \dot{\boldsymbol{\omega}}^0 = 0, \quad \boldsymbol{\beta}^0 = 0, \quad \boldsymbol{\alpha}^0 = -\mathbf{g}$

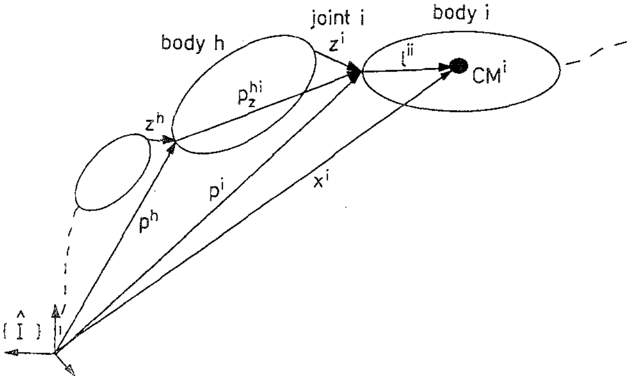


Fig. 2. Body i in a multibody structure

Recursion:

For $i = 1{:}N$

h = index of the carrying body

$$\boldsymbol{\omega}^i = \boldsymbol{\omega}^h + \boldsymbol{\Omega}^i = \boldsymbol{\omega}^h + \dot{q}^i \boldsymbol{\Phi}^i \tag{11}$$

$$\dot{\boldsymbol{\omega}}^i = \dot{\boldsymbol{\omega}}^h + \tilde{\boldsymbol{\omega}}^i \cdot \boldsymbol{\Phi}^i \dot{q}^i + \ddot{q}^i \boldsymbol{\Phi}^i \tag{12}$$

$$\boldsymbol{\beta}^i = \tilde{\dot{\boldsymbol{\omega}}}^i + \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i \tag{13}$$

$$\boldsymbol{\alpha}^i \triangleq \ddot{\mathbf{p}}^i - \mathbf{g} = \boldsymbol{\alpha}^h + \boldsymbol{\beta}^h \cdot \mathbf{p}_z^{hi} + 2\tilde{\boldsymbol{\omega}}^i \cdot \boldsymbol{\Psi}^i \dot{q}^i + \ddot{q}^i \boldsymbol{\Psi}^i \tag{14}$$

end

In order to express the mass matrix in the final form of Eq. (2), we suggest to isolate the generalized accelerations $\ddot{q}$ in this first recursion by splitting up the $\dot{\boldsymbol{\omega}}^i$, $\boldsymbol{\beta}^i$ and $\boldsymbol{\alpha}^i$ quantities as follows [7]:

$$\dot{\boldsymbol{\omega}}^i = \sum_{k \leq i} \mathbf{O}_M^{ik} \ddot{q}^k + \dot{\boldsymbol{\omega}}_C^i, \tag{15}$$

$$\boldsymbol{\beta}^i = \sum_{k \leq i} \mathbf{B}_M^{ik} \ddot{q}^k + \boldsymbol{\beta}_C^i, \tag{16}$$

$$\boldsymbol{\alpha}^i = \sum_{k \leq i} \mathbf{A}_M^{ik} \ddot{q}^k + \boldsymbol{\alpha}_C^i, \tag{17}$$

where $\sum_{k \leq i}$ represents a summation on body i and all its ancestors (i.e. belonging to the chain of bodies between i and the inertial frame).

The recursive computation of Eqs. (15)–(17) can then be done in a similar way as previously shown. This leads to the following scheme:

Initialisation:

$$\boldsymbol{\alpha}_C^0 = -\mathbf{g}; \quad \boldsymbol{\omega}^0 = \mathbf{0}; \quad \dot{\boldsymbol{\omega}}_C^0 = \mathbf{0}; \quad \boldsymbol{\beta}_C^0 = \mathbf{0}; \quad \mathbf{O}_M^{ik} = \mathbf{0}; \quad \mathbf{A}_M^{ik} = \mathbf{0} \quad i = 0{:}N,\ k > i$$

Recursion:

For $i = 1{:}N$
h = index of the carrying body

$$\boldsymbol{\omega}^i = \boldsymbol{\omega}^h + \dot{q}^i \boldsymbol{\Phi}^i \tag{18}$$

$$\dot{\boldsymbol{\omega}}_C^i = \dot{\boldsymbol{\omega}}_C^h + \tilde{\boldsymbol{\omega}}^i \cdot \boldsymbol{\Phi}^i \dot{q}^i \tag{19}$$

$$\boldsymbol{\beta}_C^i = \tilde{\boldsymbol{\omega}}^i \tilde{\boldsymbol{\omega}}^i + \tilde{\dot{\boldsymbol{\omega}}}_C^i \tag{20}$$

$$\boldsymbol{\alpha}_C^i = \boldsymbol{\alpha}_C^h + \boldsymbol{\beta}_C^h \cdot \mathbf{p}_z^{hi} + 2\tilde{\boldsymbol{\omega}}^i \cdot \boldsymbol{\Psi}^i \dot{q}^i. \tag{21}$$

For $k = 1, i$

$$\mathbf{O}_M^{ik} = \mathbf{O}_M^{hk} + \delta^{ki} \boldsymbol{\Phi}^i, \quad (\mathbf{B}_M^{ik} = \tilde{\mathbf{O}}_M^{ik}), \tag{22}$$

$$\mathbf{A}_M^{ik} = \mathbf{A}_M^{hk} + \tilde{\mathbf{O}}_M^{hk} \cdot \mathbf{p}_z^{hi} + \delta^{ki} \boldsymbol{\Psi}^i, \quad \text{with} \quad \delta^{ki} = 1 \ \text{if} \ k = i, 0 \ \text{otherwise}. \tag{23}$$

end

end

2.2 Backward dynamics

To achieve the dynamic recursion, one can easily obtain the following form for the vectorial equations of the motion of body i (see Fig. 3), on the basis of the Newton/Euler laws:

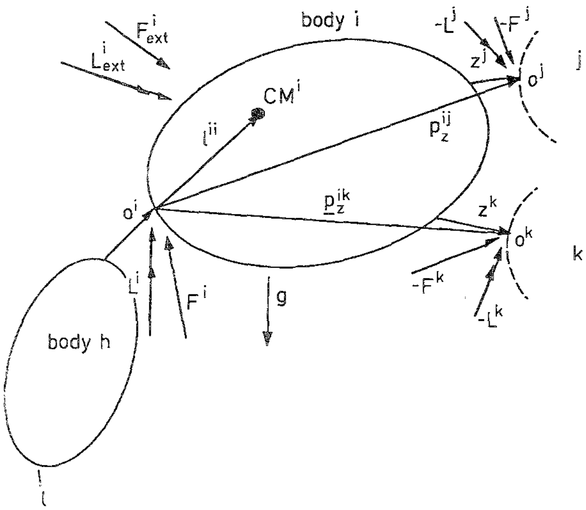


Fig. 3. Body i dynamics

The translational equation of motion of body i is

$$\mathbf{F}^i - \sum_{j \in \bar{i}} \mathbf{F}^j + \mathbf{F}^i_{\text{ext}} + m^i \mathbf{g} = m^i \ddot{\mathbf{x}}^i, \tag{24}$$

where m^i is the mass of body i, $\mathbf{F}^i$ represents the force acting on body i through the joint i, evaluated at point O^i (see Fig. 3), $\mathbf{F}^i_{\text{ext}}$ is the external force resultant applied on body i centre of mass (except gravity $\mathbf{g}$), $\sum_{j \in \bar{i}}$ denotes all the bodies "j" directly carried by i (j and k in Fig. 3).

Using Eq. (8) and the definitions (9) and (10), Eq. (24) can be rewritten as follows:

$$\mathbf{F}^i = \sum_{j \in \bar{i}} \mathbf{F}^j + \mathbf{G}^i, \quad \text{with} \quad \mathbf{G}^i \triangleq m^i(\boldsymbol{\alpha}^i + \boldsymbol{\beta}^i \cdot \mathbf{l}^{ii}) - \mathbf{F}^i_{\text{ext}}. \tag{25}$$

The rotational equation of motion of body i with respect to its centre of mass is

$$\mathbf{L}^i - \sum_{j \in \bar{i}} \mathbf{L}^j + \mathbf{L}^i_{\text{ext}} - \tilde{\mathbf{l}}^{ii} \cdot \mathbf{F}^i - \sum_{j \in \bar{i}} (\tilde{\mathbf{p}}^{ij}_z - \tilde{\mathbf{l}}^{ii}) \cdot \mathbf{F}^j = \mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \tilde{\boldsymbol{\omega}}^i \mathbf{I}^i \cdot \boldsymbol{\omega}^i, \tag{26}$$

where $\mathbf{I}^i$ is the inertia tensor of body i with respect to its centre of mass, $\mathbf{L}^i$ represents the torque acting on body i through the joint i, evaluated at point O^i, $\mathbf{L}^i_{\text{ext}}$ is the external pure torque resultant applied on body i centre of mass.

Using (25), Eq. (26) can be rewritten as follows:

$$\mathbf{L}^i = \sum_{j \in \bar{i}} \{\mathbf{L}^j + \tilde{\mathbf{p}}^{ij}_z \cdot \mathbf{F}^j\} + \tilde{\mathbf{l}}^{ii} \cdot \mathbf{G}^i - \mathbf{L}^i_{\text{ext}} + \mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}^i + \tilde{\boldsymbol{\omega}}^i \mathbf{I}^i \cdot \boldsymbol{\omega}^i. \tag{27}$$

Equations of motion (25, 27) can be recursively computed from the endbodies of the multibody system to its base. It leads to the "classical" backward recursion of the Newton/Euler scheme which provides the inverse dynamics model of the system under its implicit form (1). In order to get the mass matrix M and the C vector (Eq. 2), we also need to split up the $\mathbf{F}^i$ and $\mathbf{L}^i$ quantities to isolate the contribution of each generalized acceleration $\ddot{q}^k$

$$\mathbf{F}^i = \sum_k \mathbf{F}^{ik}_M \ddot{q}^k + \mathbf{F}^i_C, \tag{28}$$

$$\mathbf{G}^i = \sum_k \mathbf{G}^{ik}_M \ddot{q}^k + \mathbf{G}^i_C, \tag{29}$$

$$\mathbf{L}^i = \sum_k \mathbf{L}^{ik}_M \ddot{q}^k + \mathbf{L}^i_C. \tag{30}$$

These new quantities (with a low index "M" and "C") can be computed in a recursive manner by introducing relations (15, 16, 17) into the dynamic Eqs. (25, 27). One finally obtains the following scheme:

For $i = N, 1$

$$\mathbf{G}_C^i = m^i(\boldsymbol{\alpha}_C^i + \boldsymbol{\beta}_C^i \cdot \mathbf{l}^{ii}) - \mathbf{F}_{\text{ext}}^i \tag{31}$$

$$\mathbf{F}_C^i = \sum_{j \in \bar{i}} \mathbf{F}_C^j + \mathbf{G}_C^i \tag{32}$$

$$\mathbf{L}_C^i = \sum_{j \in \bar{i}} (\mathbf{L}_C^j + \tilde{\mathbf{p}}_z^{ij} \cdot \mathbf{F}_C^j) + \tilde{\mathbf{l}}^{ii} \cdot \mathbf{G}_C^i - \mathbf{L}_{\text{ext}}^i + \mathbf{I}^i \cdot \dot{\boldsymbol{\omega}}_C^i + \tilde{\boldsymbol{\omega}}^i \mathbf{I}^i \cdot \boldsymbol{\omega}^i \tag{33}$$

For $k = 1, i$

$$\mathbf{G}_M^{ik} = m^i(\mathbf{A}_M^{ik} + \tilde{\mathbf{O}}_M^{ik} \cdot \mathbf{l}^{ii}) \tag{34}$$

$$\mathbf{F}_M^{ik} = \sum_{j \in \bar{i}} \mathbf{F}_M^{jk} + \mathbf{G}_M^{ik} \tag{35}$$

$$\mathbf{L}_M^{ik} = \sum_{j \in \bar{i}} (\mathbf{L}_M^{jk} + \tilde{\mathbf{p}}_z^{ij} \cdot \mathbf{F}_M^{jk}) + \tilde{\mathbf{l}}^{ii} \cdot \mathbf{G}_M^{ik} + \mathbf{I}^i \cdot \mathbf{O}_M^{ik} \tag{36}$$

end

end

The i^{th} equation of motion of system (2) is then obtained by projecting vectorial Eqs. (32–33) and (35–36) on the i^{th} joint axis

$$C[i] = \boldsymbol{\Psi}^i \cdot \mathbf{F}_C^i + \boldsymbol{\Phi}^i \cdot \mathbf{L}_C^i \quad i = 1{:}N, \tag{37}$$

$$M[k,i] = M[i,k] = \boldsymbol{\Psi}^i \cdot \mathbf{F}_M^{ik} + \boldsymbol{\Phi}^i \cdot \mathbf{L}_M^{ik} \quad i = 1{:}N;\ k = 1{:}i. \tag{38}$$

Finally, one can write

$$\sum_k M[i,k]\,\ddot{q}^k + C[i] = Q[i]. \tag{39}$$

2.3 Comparison with an O(N) formalism

To estimate the semi-explicit Newton/Euler method within a numerical simulation context, a quantitative comparison with an O(N) formalism [14] is certainly suitable in terms of the number of arithmetic operations $(+, -, *, /)$ required for the computation of the generalized accelerations $\ddot{q}$. To achieve a consistent comparison, this implies to solve the system (2) with respect to the generalized accelerations in case of the semi-explicit Newton/Euler formalism. We perform it numerically using the Cholesky decomposition technique. The corresponding arithmetic operations have been rigourously counted up and added to those required by the generation of system (2). The following results have been obtained using the ROBOTRAN symbolic programme for both the semi-explicit Newton/Euler ("NER") and the Order N ("ODN") schemes.

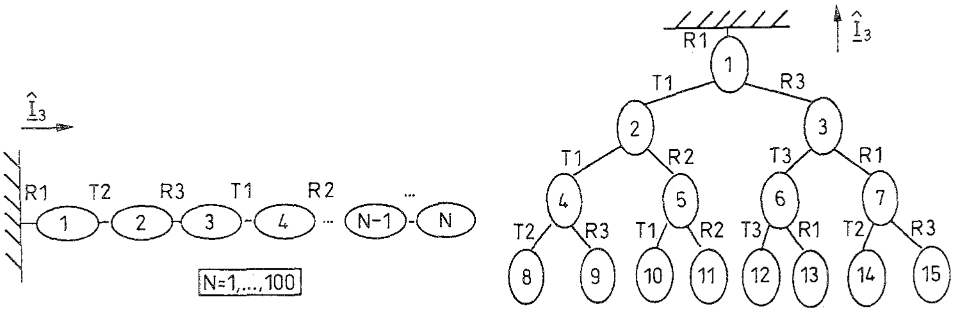
The first example consists of a "linear tree" multibody system composed of N bodies ($N = 1{:}100$) connected by one d.o.f. joints alternatively revolute ("R") and prismatic ("P") (see Fig. 4a).

The second one consists of a "binary tree" multibody system composed of N bodies ($N = 1{:}100$) in which each body carries two children bodies as shown on Fig. 4b.

Figure 5 gives the total number of arithmetical operations $(+, -, *, /)$ with respect to the number of d.o.f. (N), required by the Newton/Euler semi-explicit formalism ("NER") and by the (Order N)-formalism ("ODN").

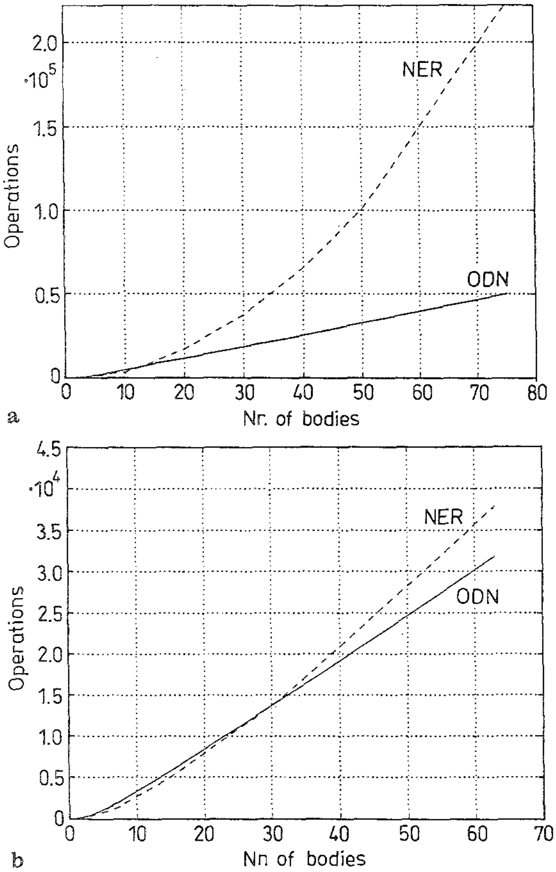
First, let N^* define the pivot number between the two formalisms as the number of d.o.f. from which the "ODN" scheme becomes more attractive than the "NER" one, in terms of arithmetical operations (see Fig. 5). This results from the linear evolution of the "ODN" scheme with respect to the number of d.o.f.

These results show that N^* is approximately equal to 12 for the linear tree, and grows to 31 for the binary tree. These values lead us to conclude that the semi-explicit Newton/Euler scheme is certainly attractive for many practical applications.



a b

Fig. 4. **a** A "linear tree" multibody system; **b** A "binary tree" multibody system



a b

Fig. 5a,b. Arithmetical operations **a** "linear tree" requirements **b** "binary tree" requirements

2.4
Constrained multibody systems

In case of kinematic constraints between the generalized coordinates of the system, $h(q) = 0$, it is easy to verify on the basis of a virtual principle that the corresponding generalized constraint forces can be written as $J(q)^T\lambda$, where $J = \partial h/\partial q^T$ is the constraints Jacobian, λ represents the Lagrange multipliers.

The system (2) becomes in that case

$$M(q)\ddot{q} + C(q, \dot{q}) = Q(q, \dot{q}) + J(q)^T\lambda, \tag{40}$$

$$h(q) = 0, \tag{41}$$

and consists of a differential/algebraic system.

As previously mentioned, we have opted for a reduction procedure leading to a purely differential system. The size of the latter is minimal, and corresponds to the number of d.o.f. of the mechanical system under consideration.

Without going into details, let summarize the main steps of the procedure, based on the well-known "coordinate partitioning method" [10].

On the basis of the set of constraints $h(q) = 0$, presumed to be independent, one can partition the generalized coordinates q into independent ("u") and dependent ("v") coordinates, and reorganize the vector q as follows:

$$q = \begin{pmatrix} u \\ v \end{pmatrix}. \tag{42}$$

The constraints and their first and second time derivatives can then be numerically solved with respect to the dependent variables

$$h(q) = 0 \Leftrightarrow v = v(u), \tag{43}$$

$$\dot{h}(q, \dot{q}) = J^v \dot{v} + J^u \dot{u} = 0 \Leftrightarrow \dot{v} = B^{vu} \dot{u}, \quad \text{with} \quad B^{vu} \triangleq -(J^v)^{-1} J^u \tag{44}$$

$$\ddot{h}(q, \dot{q}, \ddot{q}) = J^v \ddot{v} + J^u \ddot{u} + \dot{J}\dot{q} = 0 \Leftrightarrow \ddot{v} = B^{vu} \ddot{u} + b', \quad \text{with} \ b' \triangleq -(J^v)^{-1} \dot{J}\dot{q}. \tag{45}$$

Reorganizing the equations of motion (40) on the basis of the partitioning (42), and substituting $\ddot{v}$ on the basis of Eq. (45), one can easily obtain a purely differential system in terms of the independent variables u

$$\mathcal{M}(u)\ddot{u} + \mathcal{F}(u, \dot{u}) = 0, \tag{46}$$

in which both the dependent accelerations $\ddot{v}$ and the Lagrange multipliers λ have been eliminated.

Classical first- or second-order integration schemes are suitable to solve such a system. Their reliability and efficiency depends, of course, on the envisaged application.

With respect to the recursive formalisms, we have to finally point out that an O(N)-formulation usually replaces the constraints forces with equivalent external forces, leading to a more complicated procedure than the Lagrange multipliers technique. Nevertheless, we have shown in [7] that it was possible to take conjointly advantage of the O(N) philosophy and of the Coordinate partitioning method to recursively compute these unknown forces, and to specify the independent accelerations $\ddot{u}$.

On the other hand, one must recognize that the semi-explicit Newton/Euler scheme presented above, has the advantage that the constrained forces contribution is simply computed and added to the equations via the original Lagrange multipliers, leading to the well-known term $J^T\lambda$ (see Eq. (40)), easily computable.

2.5 Discussion

At this stage, we feel it relevant to point out the appeal of the previous formalism with respect to those one can find in the literature.

First of all, however, we shall emphasize that from our point of view, a comparison between formalisms with respect to the CPU time requirement is really delicate in the case of *constrained* multibody systems because:

1. one cannot dissociate anymore the formalism to generate the equations from the method to integrate them (i.e. coordinate partitioning reduction [10], constraints stabilization [15, 16] or DAE implicit solvers [8, 9]).
2. as a consequence, the CPU time should be compared on a whole simulation and, moreover, for different kinds of applications. Indeed, the "relative" performances between the methods are really problem-dependent (e.g. stiff systems).
3. among the possible methods, some of them exhibit an obvious efficiency but unfortunately lack reliability and accuracy with respect to the resolution of the non-linear algebraic constraints. In the domain of railway dynamics, this is quite out of the question, with regards to the wheel/rail contract problem [11].

In other words, we deliberately restrict the present comparison to formalisms in relative coordinates which are able to provide directly the minimal set of equations of motion of tree-like multibody systems.

It means that formalisms based on absolute coordinates are not considered here since they naturally lead on to kinematic constraints between bodies.

Among these formalisms, we unavoidably consider the O(N) approaches whose several versions have been developed by different schools (e.g. [17, 14]). That is the reason of the previous quantitative comparison (see Fig. 4). It shows that our formalism is certainly attractive for applications whose size is lower than 12, ..., 31 d.o.f., depending on the topology of the system (Fig. 5). From this point of view, notice that in most cases we systematically split up large systems (e.g. a railway vehicle) into several sub-systems (e.g. carbody, bogies), to increase both the friendliness and the efficiency of the modelling [18]. Since the order of the method is in that case a function of the size of the sub-systems, the proposed semi-explicit method really represents a competitive choice as regards the CPU time requirement.

Concerning the other formalisms in relative coordinates which also provide the equations of motion under the semi-explicit form (2), we have to point out that our formalism is purely recursive, even for the computation of the mass matrix $M(q)$. If the fully recursive nature is not to be considered as an original feature, its interest will be clearly revealed in the next section. Indeed, we have developed specific symbolic techniques to deal with recursive schemes, whatever their origin (Newton/Euler, O(N), ...), whose goal is to eliminate the useless steps and to vectorize the independent tasks.

3 Symbolic generation

The symbolic approach exhibits some substantial advantages in comparison with a pure numerical processing. Both the *legibility* and the *compactness* of the symbolic equations represent the goal of our symbolic programme ROBOTRAN [1, 7], depending on the user request. The computational efficiency being fundamental in simulation, we have decided to take conjointly advantage of the symbolic manipulation techniques, and of the recursive approaches to generate the multibody equations of motion.

We would like to emphasize that ROBOTRAN is a stand-alone symbolic programme that we have dedicated to multibody systems dynamics. This is the domain for which the programme holds a certain degree of generality. It means that symbolic manipulations are fully managed by ROBOTRAN, without any connection with a commercial symbolic package. This allows us to endow ROBOTRAN with symbolic capabilities that we couldn't have developed with a general purpose symbolic software. For example, consider the optimized symbolic generation of recursive schemes (see below) or the memory storage management during the symbolic process. The latter task, which could seem useless in view of the recent computers size and capabilities, is, however, essential in case of large symbolic computations. The ROBOTRAN memory requirements have been drastically reduced and allow us to deal with very large multibody systems (up to 100 d.o.f.) without any difficulties.

3.1 ROBOTRAN symbolic manipulations

The ROBOTRAN software (C language) is dedicated to the kinematics and the dynamics of multibody systems described in terms of relative coordinates. From the first version of the programme until now, the following symbolic manipulation procedures have been developed and included in the code [7]:

- Elimination of the "zero" quantities (zero-addition, zero-multiplication),
- Detection and simplification of redundant expressions such as "$a - a$" or "$a + b - a$", where "a" and "b" represent themselves a general expression,
- Creation of auxiliary variables to precompute quantities which occur several times in the equations,
- Trigonometric simplifications on the basis of the fundamental formulae. For example, ROBOTRAN is able to recursively simplify expressions such as $C2{*}C4{*}C56{*}C56{*}S8 + C2{*}C4{*}S56{*}S56{*}S8 + C2{*}S4{*}S56{*}C8 + S2{*}C4{*}S56{*}C8 - S2{*}S4{*}C56{*}C56{*}S8 - S2{*}S4{*}S56{*}S56{*}S8$ where CAB and SXY represent $\cos(q^A + q^B)$ and $\sin(q^X + q^Y)$, respectively. After simplifications, it simply becomes $C24{*}S8 + S24{*}S56{*}C8$.

3.2 Recursive scheme symbolic generation

Within the context of this paper, we suggest to give more details about the way ROBOTRAN generates multibody system kinematic or dynamic equations which obey a recursive structure, such as the O(N) scheme [14] or the Newton/Euler scheme in its implicit or semi-explicit form.

Indeed, one of the intrinsic characteristics of recursive schemes consists in the fact that the result of a given step of the recursion is of course a function of the previous steps results, *but not necessarily* *of all of them*, as the qualitative example illustrates in Fig. 6. The darkened elements in Fig. 6 are useless for the final results. It is thus unnecessary to compute them, and therefore to print them.

For a given multibody system, one can observe that some of the equations of a recursive scheme such as Eqs. (18–23) and (31–38) in their scalar form are useless because they don't contribute towards the final results, i.e. the "last" recursion steps such as Eqs. (37, 38) denoting the mass matrix $M(q)$ and the vector $C(q, \dot{q})$ of system (2).

To detect the useless steps for a given multibody model, all the recursive (useful and useless) equations are first stored by ROBOTRAN inside a double-linked list, during the symbolic generation. Before printing the final equations, the list is covered via C-pointers from the end (results) to the beginning (data), to check and mark dependencies, and to cancel the useless elements (i.e. equations) whose printing can be finally avoided.

The saving is "application-dependent" but can reach 30% in terms of arithmetical operations. This cannot be neglected.

Another interesting characteristic of a recursive scheme, from which it is possible to profit via a symbolic programme, is that the results of a given step of the recursion are a function of the previous steps results, *but not necessarily of the last one(s)*, as shown on the example given in Fig. 7. One can easily observe that $A2$ needn't $A1$ to be computed as well as $C2$ needn't $C1$. One can then imagine to compute them in parallel ($A1$ and $A2$, $C1$ and $C2$), leading to a so-called "vectorized form" of the recursive scheme.

The automation of such a "vectorization process" is within ROBOTRAN's capabilities in the same way as the detection of useless equations. Moreover, the process is optimal in the sense that, for a given equation of the serial scheme (e.g. "$C2 = B + B2$" in the previous example), ROBOTRAN assigns to the left hand side a vectorial step index "VSI", defined as the number of the step in the vectorized form $VSI(C1) = VSI(C2) = 3$, $VSI(R) = 5$ in the example

$$VSI(\text{left-hand side}) = 1 + \max_j \{VSI(j^{\text{th}} \text{ term of right hand side})\}.$$

In the example, $VSI(B) = 0$ and $VSI(B2) = 2$ being previously computed, it gives: $VSI(C2) = 3$.

In this way, one can ensure that the number of vectorial steps is minimal. Moreover, the method is fully independent of the recursive scheme type. Indeed the vectorization process is purely of a symbolic nature.

In practice, one observes that the reduction of the number of steps between a serial and a vectorized recursive scheme is really amazing. Using, for instance, the semi-explicit Newton/Euler scheme, ROBOTRAN generates 18 vectorial steps against 560 serial steps for a 3D pendulum (9 d.o.f.), and 23 vectorial steps against 1448 serial steps for a railway bogie (22 d.o.f.).

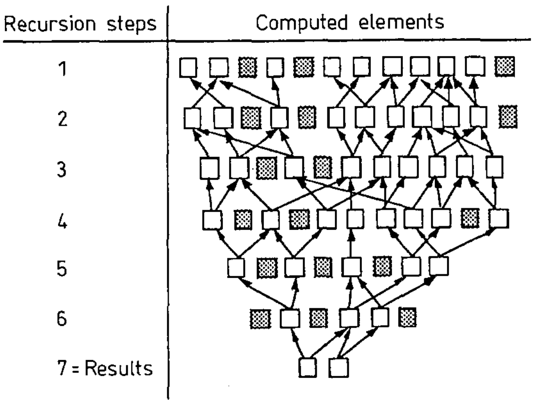


Fig. 6. A qualitative recursive scheme

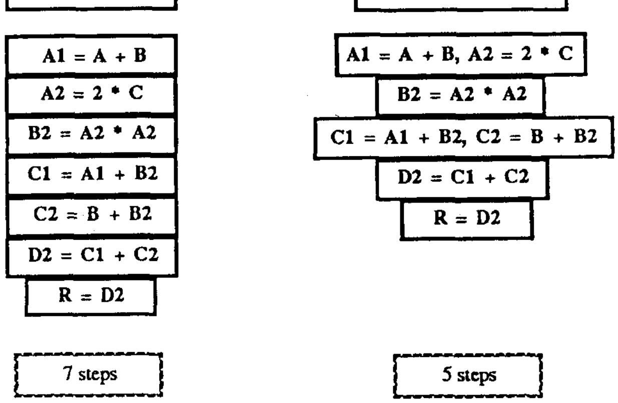


Fig. 7. Recursive scheme vectorization

To avoid any confusion, we should emphasize that the ROBOTRAN vectorization doesn't reduce the number of arithmetic operations, but only reorganizes the recursive equations in a vectorized form. We only hope that the latter form will be advantageously exploited in vectorial computer architectures in the future.

We already profit from this new process, since ROBOTRAN is able to generate the equations in the MATLAB syntax. Indeed, the latter programme reaches its best performances with vectorized instructions. To quantify this for a medium-size multibody model, a railway bogie, we obtained a time reduction factor of about 2.8.

4
Application in railway dynamics

All these developements have been applied to a large number of railway vehicles, and in particular to non-conventional railway bogies designed by the Belgian company B.N.-Eurorail. We have particularly focused our attention on the so-called BAS 2000 bogie (Fig. 1, [19]), which consists of a complex articulated mechanism: 13 bodies connected by revolute or spherical joints, 6 three-dimensional kinematic loops, carried by four independent wheels via a vertical primary suspension.

From a modelisation point of view, the bogie represents (see Fig. 8): 24 relative variables linked by 12 kinematic constraints leading to 12 d.o.f.

Such an application is really enriching from a multibody point of view. Both the kinematics and the dynamics of the bogie have been generated by ROBOTRAN using recursive formulations. As regards the wheel/rail contact problem, we had to develop a new geometrical contact model for independent wheels [11], since classical models deal only with rigid wheelsets [20].

4.1
A typical numerical example

In accordance with the Belgian company B.N.-Eurorail demands, several numerical treatments and analyses (quasi-static equilibrium, straight track modal analysis, non-linear simulation) have been performed on different configurations of the BAS 2000 bogie. Here below, a non-linear simulation is proposed and is related to the bogie dynamic behaviour on a pure straight track. The model, Fig. 9, consists of a BAS 2000 bogie carrying half a carbody, whose articulation with the previous one (point P) is assumed to move perfectly at constant speed along the track centre.

Figure 11 represents the evolution of the A-carbody yaw angle (Ψ) with respect to time, and points out a transversal "rebounding" phenomenon of the bogie, strongly influenced by the wheel/rail flanges gauge (Fig. 10).

One observes that for a small gauge (2 mm on the Figure), the rebounding effect disappears. The bogie then leans continuously against one of the rails (the left one in the present case).

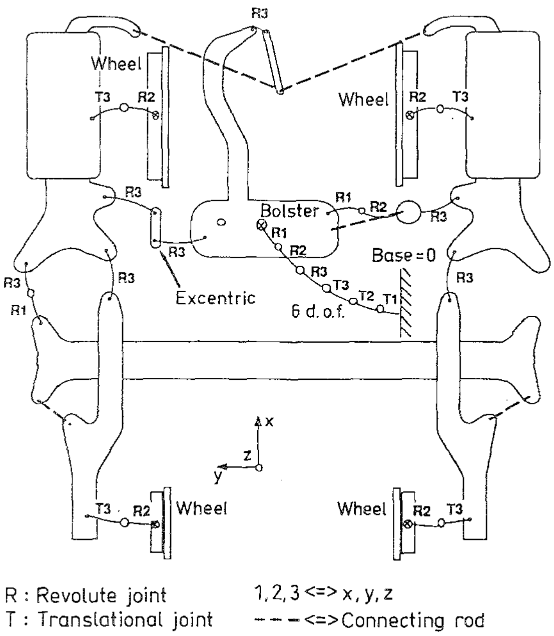


Fig. 8. BAS 2000 bogie ROBOTRAN model

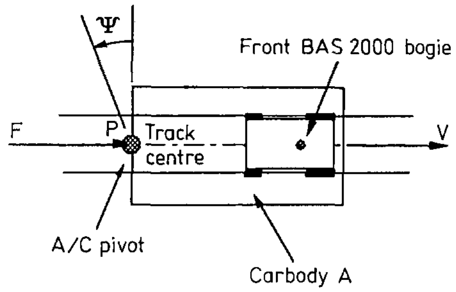


Fig. 9. BAS 2000 model

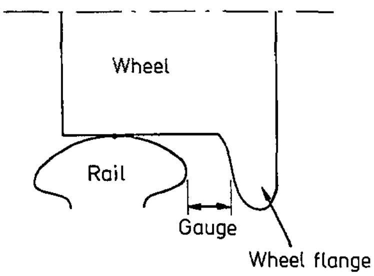


Fig. 10. Wheel/rail transversal gauge

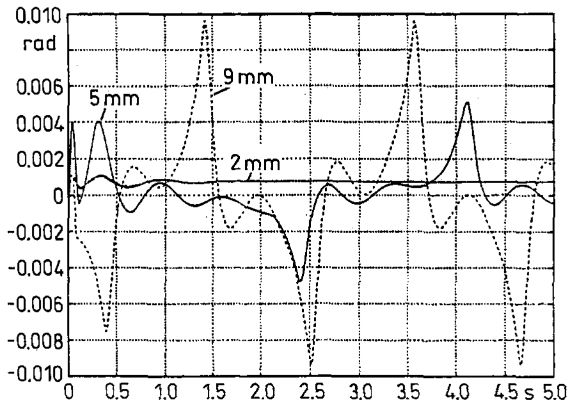


Fig. 11. BAS 2000 dynamic behaviour on a straight track

4.2
Computational efficiency

The main motivation of the previous developments (recursive multibody formalism in symbolic form) was to reduce the CPU time of the numerical simulation of bogies and tramways for the Belgian B. N. company. Indeed, the BAS 2000 bogie is part of a whole tramway, consisting of three carbodies carried by three bogies. With the vehicle, whose complexity arises from its size (51 bodies, 20 loop constraints) and from the wheel/rail contact (12 independent wheels), a simulation using a non-recursive formalism based on the d'Alembert Power Principle and implemented numerically [21] was next to impossible in practice.

The reformulation of the latter formalism under a symbolic form was the first step. A time saving factor between 5 and 6 has been found with respect to the previous approach. Notice that not only the dynamics but also the loop constraints, their time derivatives and the wheel kinematics are generated symbolically. This represents the major part of the model.

As regards the semi-explicit recursive scheme, its first appeal was to allow the full symbolic generation of very large multibody systems without any difficulties. Secondly, it has allowed to reduce the CPU time by a factor 4 to 5 with respect to the non-recursive method for the simulation of the BAS 2000 bogie. This saving factor is clearly problem dependent.

Altogether, these improvements have allowed to simulate the entry curving of a whole tramway with a full non-linear geometrical model for the wheel/rail contact in less than three hours on a 1.4 M flops SUN Station.

5
Conclusions and prospects

The more complex and sophisticated are the mechanical systems to analyse, the more the reliability and the efficiency of the mathematical model have to be increased.

From the efficiency point of view, we have particularly looked into the problem of the equations generation by developing a modified recursive Newton/Euler scheme to minimize the arithmetic operations cost.

As regards the symbolic generation, we have shown that the latter technique was really suitable in case of recursive schemes and exhibited some amazing capabilities to vectorize the equations of motion, independently of the envisaged scheme.

A very enriching application, the BAS 2000 articulated bogie, has allowed us to successfully apply these developments and convinced us that the limitation of the symbolic computation for large models was definitively surmounted.

Future developments will be related to:

- the use of the symbolic approach in case of multibody systems with flexible bodies as proposed in [22],
- the implementation of implicit integration schemes in conjunction with the symbolic approach; the latter indeed can be usefull in computing analytically and recursively the tangent matrices of the linearized system (the formalism presented above already computes the mass matrix),
- the use of the numerical parallel computation in case of large multibody models when split up into several sub-systems.

References

1. **Maes, P.; Samin, J. C.; Willems, P. Y.:** ROBOTRAN. In: Schiehlen, W. (ed.) Multibody System Bandbook, pp. 225–245. Berlin: Springer 1989
2. **Andrzejewski, Th.; Bock, H. G.; Eich, E.; von Schwerin, R.:** Recent advances in the numerical integration of multibody systems. In: Schiehlen, W. (ed.) Advanced Multibody System Dynamics, pp. 49–66. Dordrecht: Kluwer Aademic Publishers 1993
3. **Wittenburg, J.:** Dynamics of systems of rigid bodies. Stuttgart: Teubner 1977
4. **Kane, Th. R.; Levinson, D. A.:** Dynamics: Theory and applications. New-York: McGraw-Hill 1985
5. **Roberson, R. E.; Schwertassek, R.:** Dynamics of multibody systems. Berlin: Springer 1988
6. **Eichberger, A.; Führer, C.; Schwertassek, R.:** The benefits of parallel multibody simulation and its application to vehicle dynamics. In: Schiehlen, W. (ed.). Advanced Multibody System Dynamics, pp. 107–126. Dordrecht: Kluwer Academic Publishers 1993
7. **Fisette, P.:** Génération symbolique des équations du mouvement de systèmes multicorps et application dans le domaine ferroviaire. PHD Thesis, University of Louvain-la-Neuve, Belgium, 1994
8. **Petzold, L.:** Methods and softwares for differential/algebraic systems. In: Haug, E. J.; Roderic, C. D. (eds.) NATO ASI Series, Serie F: Computer and Systems Sciences 69, pp. 127–140. Berlin: Springer 1989
9. **Führer, C.; Leimkuhler, B.:** A new class of generalized inverses for the solution of discretized Euler-Lagrange equations. In: Haug, E. J.; Roderic, C. D. (eds.) NATO ASI Series, Serie F: Computer and Systems Sciences 69, pp. 143–154. Berlin: Springer 1989
10. **Wehage, R. A.; Hung, E. J.:** Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. J. Mech. Design 134 (1982) 247–255
11. **Fisette, P.; Samin, J. C.:** A new wheel/rail contact model for independent wheels. Arch. Appl. Mech. 64 (1994) 192–205
12. **Renaud, M.:** Quasi-minimal computation of the dynamic model of a robot manipulator utilizing the Newton-Euler formalism and the notion of augmented body. In: Proc. IEEE Int. Conf. Robotics and Automation, Raleigh, North Carolina, 1987, pp. 1677–1682
13. **Luh, J. Y. S.; Walker, N. W.; Paul, R. P. C.:** On-line computational scheme for mechanical manipulators. Trans. ASME/J. Dyn. Syst. Meas. Control 102 (1980) 69–76
14. **Schwertassek, R.; Rulka, W.:** Aspects of efficient and reliable multibody systems simulation. In: Haug, E. J.; Roderic, C. D. eds. NATO ASI Series, Serie F: Computer and Systems Sciences 69, pp. 55–96. Berlin: Springer 1989
15. **Baumgarte, J.:** Stabilization of constraints and integrals of motion computer methods. Applied Mechanics and Engineering 1 (1972) 1–16
16. **Bae, D. S.; Yang, S. M.:** A stabilization method for kinematic and kinetic constraint equations. In: Haug, E. J.; Roderic, C. D. eds. NATO ASI Series, Serie F: Computer and Systems Sciences 69, pp. 209–232. Berlin: Springer 1989
17. **Bae, D. S.; Haug, E. J.:** A recursive formulation for constrained mechanical systems, part 1: Open loop. Mech. of Structures and Machines 15 (1987) 359–382
18. **Fisette, P.; Lipinski, K.; Samin, J. C.:** Symbolic generation of large degrees of freedom multibody systems. Proc. 6th DYNAME 95 Symposium, Brazil, 6–10 march 1995, pp. 299–302
19. **Fisette, P.; Samin, J. C.:** Lateral dynamics of a light railway vehicle with independent wheels. Vehicle System Dynamics 20 (1991) 157–171
20. **Kortüm, W.; Sharp, R. S.:** Multibody computer codes in vehicle system dynamics (supplement to Vehicle System Dynamics 22). Amsterdam/Lisse: Swets & Zeitlinger 1993
21. **Maes, P.; Samin, J. C.; Willems, P. Y.:** Autodyn. In: Schiehlen, W. (ed.) Multibody System Handbook, pp. 225–245. Berlin: Springer 1989
22. **Fisette, P.; Samin, J. C.; Willems, P. Y.:** Contribution to symbolic analysis of deformable multibody systems. Int. J. Num. Meth. Eng. 32 (1991) 1621–1635