THE ARCHIVE OF MECHANICAL ENGINEERING

VOL. LVII 2010 Number 2

10.2478/v10180-010-0011-1

Key words: multibody systems, sensitivity analysis, symbolic generation, vehicle dynamics

ANTOINE PONCELET[*], *JEAN-FRANÇOIS COLLARD*[*], *PAUL FISETTE* [*]

CONTRIBUTION OF SYMBOLIC GENERATION TO THE SENSITIVITY ANALYSIS OF AUTOMOTIVE SUSPENSION PARAMETERS

The purpose of the present research relates to the sensitivity analysis of road vehicle comfort and handling performances with respect to suspension technological parameters. The envisaged suspension being of semi-active nature, this implies first to consider an hybrid modeling approach consisting of a 3D multibody model of the full car – an Audi A6 in our case – coupled with the electro-hydraulic model of the suspension dampers. Concerning parameter sensitivitie, the goal is to capture them for themselves – and not necessarily for optimization purpose – because their knowledge is of a great interest for the damper manufacturer.

An important issue of the research is to consider objective functions which are based on complete time integrations along a given trajectory, the goal being – for instance – to quantify the sensitivity of the carbody rms acceleration (comfort) or of the vehicle overturning character (handling) with respect to suspension parameters. On one hand, the accuracy of the various partial derivatives computation can be greatly enhanced thanks to the symbolic capabilities of our ROBOTRAN multibody program. On the other hand, the computational efficiency of the process also takes advantage of the recursive formulation of the multibody equations of motion which must be time integrated with respect to both the generalized coordinates and their partial derivatives in case of the so-called direct method underlying sensitivity analysis.

1. Introduction

Research context

Within the framework of our collaboration with the Tenneco Automotive Company which develops car suspension systems (Monroe™ dampers), it recently appeared that the role or – more specifically – the precise impact of

[*] *Center for Research in Mechatronics, Université catholique de Louvain, Place du Levant 2, 1348 Louvain-la-Neuve, Belgium. E-mail: antoine.poncelet@uclouvain.be; web page: http://www.cerem.be, E-mails: jf.collard@uclouvain.be, paul.fisette@uclouvain.be*

suspension parameters on comfort and handling performances needed to be better quantified. For instance, in the course of a recent multibody modeling project with Tenneco, it was shown by simulation that, among the 21 (!) suspension parameters they have to deal with, some of them have – finally – no (or very little) influence on specific dynamic performances of the car (an Audi A6 in this case): this non-sensitive nature was, surprisingly for us, a very fruitful result for them. As another more recent example which relates to human subjective evaluation of car performances, it is obviously an accepted fact that the increase of the carbody roll motion has a negative effect, while a decrease does not necessarily have a positive one! This "unilateral" observation – which results only from human perceptions – allows us to illustrate the context and the purpose of the present research: the development of a precise, robust and efficient method to quantify suspension parameter sensitivity for given parameter ranges and specific car trajectories and excitations.

Problem formulation

The applications to be dealt with consist of modern cars with multi-link suspensions equipped with various possible morphologies of passive or semi-active hydraulic dampers. Considering the complexity of such systems and the fact that model validation is mandatory in such a collaborative research (see Fig. 1 for example), the model must be sufficiently refined and accurate. It consists of a 3D multibody model of the car, which includes multi-link suspension kinematics, wheel/ground force models issuing from [5], and a dynamic model of the suspension's hydraulic system [6] taking oil compressibility into account (see Fig. 2). Such a model and its experimental validation have previously been presented [8], the emphasis having been placed on the way to obtain an hybrid multibody-hydraulic model coupled at the equational level, without resorting to co-simulation techniques.

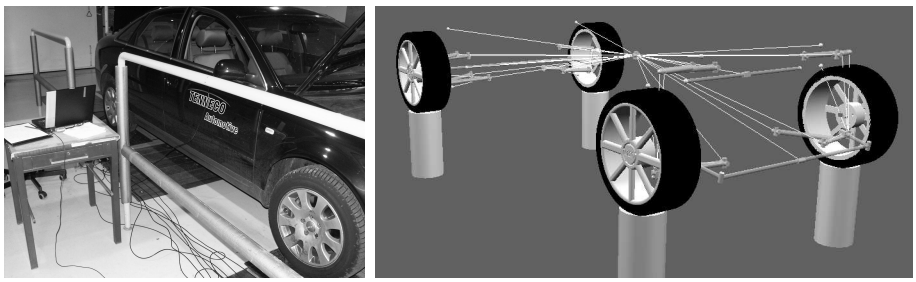

Fig. 1. Model validation - left: Audi A6 on the shaker, right: the MBS simulation (wireframe snapshot)

For the present project, the relevant sensitivity analyses are already well defined in terms of parameters p and objective functions f to be considered: typically the former (p) denote suspension design parameters (orifice section, pipe length, electro-valve current etc.); the latter ($\psi(p,t)$) mainly concern rms – possibly filtered – accelerations, mean dissipated power, carbody yaw or roll rates etc., thus requiring a time integration from $t^0 \rightarrow t$ along prescribed trajectories (combining straight/curved, smooth/uneven stretches of road, etc.).

Further to our previous experience in vehicle dynamics, particular attention will be paid to the non-linear character of the multibody-hydraulic equations, with respect to both the generalized coordinates q and the envisaged parameters p, and to the computational cost of the objective function sensitivity $\frac{\partial \psi(p,t)}{\partial p}$. Indeed, the latter depends on full trajectories (from t^0 to t) and must be computed, not only around a nominal parameter value p_k, but for a range $p_{inf} < p_k < p_{sup}$, that is: $\frac{\partial \psi(p,t)}{\partial p}|_{p=p_k}, k = 1...n$.

Rather than investigating new sensitivity methods, the emphasis will primarily be placed here on the *problem formulation*, by tackling numerical/symbolic differentiation, time integrator accuracy and computer efficiency, within the framework of the so-called *direct* method [4].

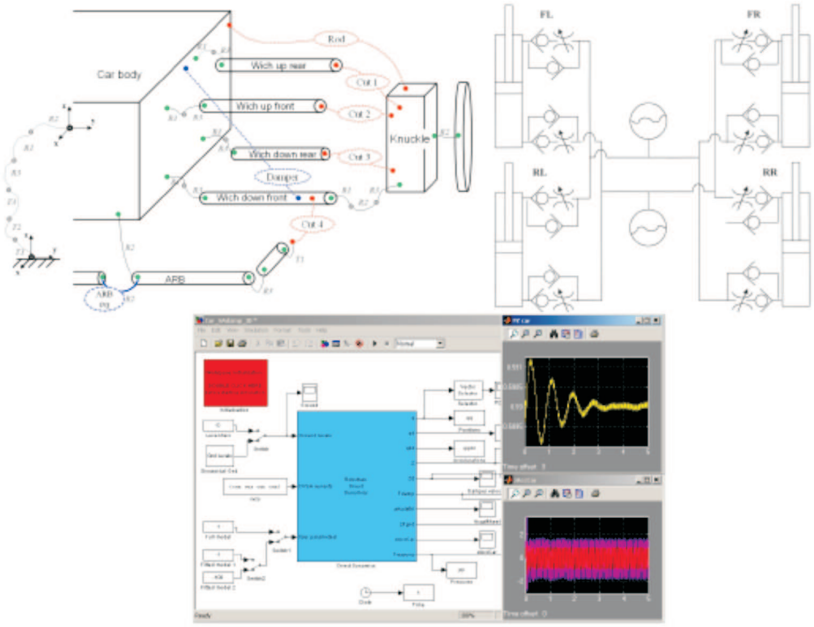

Fig. 2. Audi A6: ROBOTRAN multibody-hydraulic model and time simulation

2. Sensitivity analysis: formulation

In our case, we formulate the multibody dynamic equations in terms of relative joint coordinates q for which possible constraints must be satisfied at any time, as for instance to ensure body loop closure (ex.: car multi-link suspension mechanisms). The general formulation of the equations of motion can be written as:

$$M(q,\delta)\,\ddot{q} + c(q,\dot{q},f,\delta) = J^t\lambda \tag{1}$$

$$h(q) = 0 \tag{2}$$

$$\dot{h}(q,\dot{q}) = J(q)\dot{q} = 0 \tag{3}$$

$$\ddot{h}(q,\dot{q},\ddot{q}) = J(q)\ddot{q} + \dot{J}\dot{q}(q,\dot{q}) = 0 \tag{4}$$

where:

- M is the generalized mass matrix of the unconstrained system;
- δ gathers the system parameters (dimensions, mass, inertia, etc.) together;
- $f(q,\dot{q},\delta)$ represents all the – internal and external – contributive forces and torques acting on the system;
- $J \triangleq \dfrac{\partial h}{\partial q^t}$ denotes the constraint Jacobian matrix;
- $\dot{J}\dot{q}(q,\dot{q})$ is the quadratic term (expression in $\dot{q}^i\dot{q}^j$) of the constraints at acceleration level;
- λ represents the Lagrange multipliers associated with the constraints.

As far as we are concerned, we have definitively opted for the Coordinate Partitioning technique [1] in which the joint coordinates q – as well as the constraint Jacobian matrix J – are partitioned as follows:

$$q = \begin{pmatrix} u \\ v \end{pmatrix} \;;\; J = \begin{pmatrix} J_u \\ J_v \end{pmatrix} \tag{5}$$

where u denotes the subset of *independent* coordinates and v the subset of *dependent* coordinates. Based on this partitioning, it is possible to solve the constraints at position*, velocity and acceleration levels,

$$v = v(u) \;;\; \dot{v} = -J_v^{-1}\,J_u\,\dot{u} \;;\; \ddot{v} = -J_v^{-1}\left(J_u\ddot{u} + \dot{J}\dot{q}\right) \tag{6}$$

and to reduce the DAE system (1 – 4) into a pure ODE system [7] witch can be synthetically written:

$$M_r(u,\delta)\ddot{u} + F_r(u,\dot{u},f,\delta) = 0 \tag{7}$$

* using an iterative Newton-Raphson algorithm: $v^{k+1} = v^k + \ldots$

This ODE system constitutes the equations of motion of the constrained multibody system described in terms of the independent generalized coordinates u. Finally, by solving this linear system (using for instance a Cholesky decomposition of the reduced mass matrix M_r), the independent accelerations $\ddot{u}$ can be explicitly computed:

$$\ddot{u} = \phi(u, \dot{u}, f, \delta) \tag{8}$$

At this point, it is worth mentioning that for any MBS containing nonlinear constraints (ex. resulting from closed-loops), ROBOTRAN is able to generate the explicit reduced form (8) in a *fully symbolic* way, including the loop closure and the reduction process [2].
Thus for the explicit form (8), a *unique* function (in C, Matlab, ect.), generated by ROBOTRAN, computes the accelerations $\ddot{u}$ (and possible "subproducts" such as the Lagrange multipliers λ and the dependent coordinates $v, \dot{v}, \ddot{v}$) according to Fig. 3. The input of such functions are:

- u, $\dot{u}$: the generalized position and velocities (from the previous time step for instance);
- δ, the system parameters;
- p, any parameter (or set of parameters) with respect to which a sensitivity analysis is envisaged.

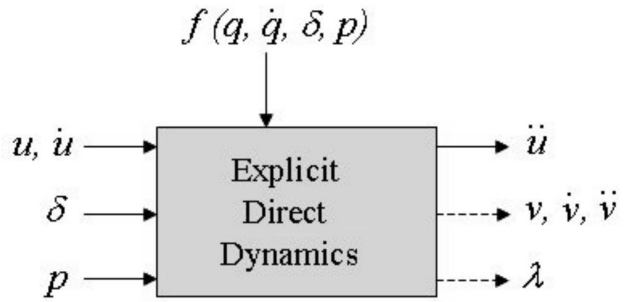

Fig. 3. Symbolic computation of the direct dynamics for constrained MBS

Let us emphasize that from a pure computational point of view, the availability of the explicit direct dynamics in symbolic form (based on recursive formalisms) allows us to simulate *better than real time* full car models with wheel/ground interactions and multi-link suspensions, in the SIMULINK environment. This efficiency, mainly due to the simplification capabilities of the ROBOTRON symbolic generator, obviously represents an interesting "starting point" for the proposed sensitivity process, particularly costly computationally speaking.

In the multibody dynamic context, a typical objective function $\psi(p)$ for sensitivity analysis – or optimization – purpose (see [4], [3]) can be written as:

$$\psi(p) = G^1(t^1, u^1, \dot{u}^1, p) + \int_{t^0}^{t^1} F(t, u, \dot{u}, \ddot{u}, p)\, dt \tag{9}$$

in which:

- t^0 and t^1 are the initial and final simulation time;
- G^1 refers to the final state (ex. the configuration at t^1 of a given body of the system);
- F depends on the dynamic behavior of the system in the time interval $[t^0, t^1]$, such as the rms vertical acceleration of the car driver, the mean power dissipated at the wheel/ground contact, etc.

For the general objective function (9), sensitivity analysis consists in computing $\dfrac{d\psi}{dp}$, that is:

$$\frac{d\psi}{dp} = \frac{\partial G^1}{\partial u^1}.\frac{du}{dp}\bigg|_{t^1} + \frac{\partial G^1}{\partial \dot{u}^1}.\frac{d\dot{u}}{dp}\bigg|_{t^1} + \frac{\partial G^1}{\partial p} + \int_{t^0}^{t^1} \left(\frac{\partial F}{\partial u}.\frac{du}{dp} + \frac{\partial F}{\partial \dot{u}}.\frac{d\dot{u}}{dp} + \frac{\partial F}{\partial \ddot{u}}.\frac{d\ddot{u}}{dp} + \frac{\partial F}{\partial p} \right) dt \tag{10}$$

Let us first point out that in our case the only state variables are the *independent* coordinates u (and $\dot{u}$). Indeed, the remaining variables v (and $\dot{v}$) have been eliminated from the model during the reduction process (from the DAE (1 – 4) to the ODE (7) or (8)), v and $\dot{v}$ being expressed in terms of u and $\dot{u}$ according to the constraints solution.

Within expression (10), the unknown sensitivity matrices $\dfrac{du}{dp}(t)$, $\dfrac{d\dot{u}}{dp}(t)$ and $\dfrac{d\ddot{u}}{dp}(t)$ can be computed via the so-called *direct method* [4] which consists in solving the differential equations for sensitivity matrices *simultaneously* with the equations of motion as explained here below.

Considering the semi-explicit form of the dynamic equations (7), any δp yields a small change in the system response δu, $\delta \dot{u}$ and $\delta \ddot{u}$ and thus sensitivity matrices can be calculated via the following equations in which, for sake of simplicity, we have defined $\Gamma(u, \dot{u}, \ddot{u}, \delta) \triangleq M_r(u, \delta)\ddot{u} + F_r(u, \dot{u}, f, p)$:

$$M_r.\frac{d\ddot{u}}{dp} + \frac{\partial \Gamma}{\partial \dot{u}}.\frac{d\dot{u}}{dp} + \frac{\partial \Gamma}{\partial u}.\frac{du}{dp} + \frac{\partial \Gamma}{\partial p} = 0 \tag{11}$$

This equation can be time integrated simultaneously with the equations of motion (7). When using first order time integration schemes, second order

derivatives $\ddot{u}$ and $\dfrac{d\ddot{u}}{dp}$ can be gathered together to define a new auxiliary variable y:

$$y = \begin{pmatrix} y^1 \\ y^2 \end{pmatrix} \triangleq \begin{pmatrix} u \\ \dot{u} \end{pmatrix}; \dot{y} = \begin{pmatrix} \dot{y}^1 \\ \dot{y}^2 \end{pmatrix} \triangleq \begin{pmatrix} \dot{u} \\ \ddot{u} \end{pmatrix};$$

and similarly for the sensitivities u_p:

$$y_p = \begin{pmatrix} y_p^1 \\ y_p^2 \end{pmatrix} \triangleq \begin{pmatrix} \dfrac{du}{dp} \\ \dfrac{d\dot{u}}{dp} \end{pmatrix} \; ; \; \dot{y}_p = \begin{pmatrix} \dot{y}_p^1 \\ \dot{y}_p^2 \end{pmatrix} \triangleq \begin{pmatrix} \dfrac{d\dot{u}}{dp} \\ \dfrac{d\ddot{u}}{dp} \end{pmatrix}$$

The global system reads:

$$\begin{aligned} \dot{y}^1 &= y^2 \\ \dot{y}^2 &= -M_r^{-1}(y^1, p)\left(F_r(y^1, y^2, f, p)\right) \\ \dot{y}_p^1 &= y_p^2 \\ \dot{y}_p^2 &= -M_r^{-1}(y^1, p)\left(\frac{\partial \Gamma}{\partial \ddot{u}} . \frac{dy_p^2}{dp} + \frac{\partial \Gamma}{\partial u} . \frac{dy_p^1}{dp} + \frac{\partial \Gamma}{\partial p}\right) \end{aligned} \tag{12}$$

which has the following suitable form for first order integrators

$$\dot{z} = \phi(z, t) \;\text{ where }\; z \triangleq \begin{pmatrix} y \\ y_p \end{pmatrix}$$

As previously mentioned, the reduced mass matrix inversion in (12_b, 12_d) will be advantageously replaced by a Cholesky decomposition process.
The computational cost of the equations of motion (12_b) can be noticeably reduced by means of the symbolic generation underlying the ROBOTRAN program (factor 8 to 10) and, additionally, by using the explicit formulation (8) (fully symbolic generation) instead of the semi-explicit one (7) (factor 2 to 4).

The main reason of the "symbolic versus numerical" benefit comes more from the *recursive* nature of the dynamic equations which is at the root of the recursive symbolic elimination of useless equations in ROBOTRAN (see Section 3.2), than from the symbolic simplification of expressions, obviously also performed by the program.

3. Sensitivity analysis: symbolic computation

3.1. Introduction

Considering now the sensitivity equations $(12)_d$, our goal is to generate them also symbolically. First of all, it is worth noting that the complexity of these equations mainly comes from the computation of the partial derivatives $\frac{\partial \Gamma}{\partial \ddot{u}}$ and $\frac{\partial \Gamma}{\partial u}$ for large multibody systems like those we have to deal with (full car with 3D suspensions, full railway vehicles, etc.) and for which the direct dynamics $(12)_b$ is far from being trivial for symbolic programs. Moreover, in the context of the Coordinate Partitoning process, the computation of equations $(12)_d$ is also not trivial and the associated complexity strongly depends on the availability of the various elements; in particular, if the constraints $h(q)$ are solved at position, velocity levels in distinct processes or "subroutines" according to equations (6), the partial derivatives of the dependent coordinates $\frac{dv}{dp}$ and $\frac{d\dot{v}}{dp}$ will be needed *explicitly*, as well as those related to the constraints and the Jacobian themselves, for the computation of $\frac{\partial \Gamma}{\partial \dot{u}}$ and $\frac{\partial \Gamma}{\partial u}$.
At position level for instance, we have to compute

$$\frac{dv}{dp} = -J_v^{-1} \frac{\partial h}{\partial p}$$

which requires the explicit computation of the constraints sensitivity $\frac{\partial h}{\partial p}$. The same conclusion holds for constraints derivatives.
For these reasons, the *explicit* fully symbolic formulation (8) of the direct dynamics (see Fig. 3), is perfectly suited to a recursive symbolic differentiation process directly applied to the output $\ddot{u}$, to produce $\frac{d\ddot{u}}{\partial p}$ straightforwardly. This is the purpose of the two next Subsections.

3.2. Recursive symbolic generation in ROBOTRAN

To clarify the following explanations, we name a *recursive scheme* denotes any formalism (kinematic, dynamic, direct, inverse, ...) written as one or more algorithmic loops covering the kinematic chains of bodies. Typically, the well-known Newton/Euler recursive formalism, which underlies the ROBOTRAN symbolic generation of system (1) for unconstrained systems,

represents a recursive scheme consisting of two algorithmic loops: one for the forward kinematics, the second for the backward dynamics [7].
To illusttrate the recursive nature of these formalisms, let us consider, in the forward kinematics, the relation which expresses the absolut angular velocity of a given body 3 with respect to body 2, its parent body in the chain:

$$\boldsymbol{\omega}^3 = \boldsymbol{\omega}^2 + \boldsymbol{\phi}^3 \dot{q}^3 \tag{13}$$

in which joint 3 is revolute along unit vector $\boldsymbol{\phi}^3$. The matrix form of this equation symbolically reads in the body 3 attached-frame $\{\hat{\mathbf{X}}^3\}$:

$$\begin{aligned} \text{OM13} &= \text{qd(3)+ OM12} \\ \text{OM23} &= \text{OM22*C3+OM32*S3} \\ \text{OM33} &= \text{-OM22*S3+OM32*C3} \end{aligned} \tag{14}$$

where OMij denotes the i^{th} component of the j^{th} body angular velocity in frame $\{\hat{\mathbf{X}}^j\}$, qd(j) denotes $\dot{q}^j$ and Cj, Sj represent $\cos(q^j)$, $\sin(q^j)$ respectively.

In ROBOTRAN each recursive equation of type (15), once symbolically evaluated, is stored in a dynamic list linked by C-pointers as indicated in Fig. 4. By "symbolically evaluated", we mean that the right hand side (RHS) of each equation is a symbolic expression – previously simplified – stored in the computer memory and located via C-pointers.

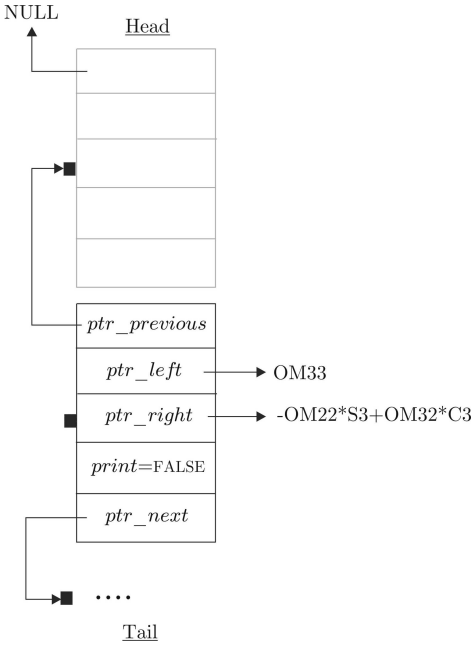

Fig. 4. C linked-list of recursive equations

The process can be summarized in two tips:

1. A recursive equation LHS=RHS (ex.: OM33= −OM22*S3+OM32*C3) is symbolically evaluated. The LHS (OM33) becomes a new *auxiliary* variable, i.e. an elementary expression, which is generated as soon as the RHS is evaluated.
2. The list is dynamically increased by one unit in which:
   - pointer *ptr_previous* links the unit to the previous one in the recursive algorithm (*ptr_previous* = NULL for the first element),
   - pointer *ptr_next* will link the unit to the next one (*ptr_next* = NULL for the last element),
   - pointer ptr_left points to the LHS leaf expression,
   - pointer ptr_right points to the RHS expression,
   - The boolean flag *print* is a priori set to FALSE because the corresponding expression do not a priori contribute to the expected results.

Elimination process

The linked list is complete as soon as all the final results have been symbolically evaluated (ex.: M, c in equation (1) or $\ddot{u}$ in equation (8)).
Each result (ex. for the mass matrix: $M_{(i,j)}$, $i = 1 : n$, $j = i : n$) having the same structure as the previous recursive equations (i.e. LHS = RHS) is also stored in the list, but of course with a *print* flag set to TRUE. Once they have been evaluated, the selection process can start by scanning the list from tail to head, and can be summarized as follows:

1. Start from the tail element of the list (i.e. the last symbolically evaluated equation (ex.: the first mass matrix element $M_{(1,1)}$),
2. Analyze the contents of the equation:
   - if the corresponding *print* flag is TRUE, then for each auxiliary variable encountered in the RHS of the equation, find and mark, by setting *print* to TRUE, the recursive equation in the list which symbolically evaluates this variable,
   - else, go directly to step 3, because the corresponding equation is superfluous,
3. Go to the previous element of the list whose address is given by the pointer *ptr_previous*:
   - if the address is the NULL pointer, the head has been reached and the process is finished: go to step 4,
   - else, return to step 2,
4. Print in the output file, by covering the list from head to tail, the equations whose *print* flag has been set TRUE.

Figure 5 summarizes this process using an academic example.

Thanks to this recursive simplification process, we note that more than 30% of the equations are superfluous (for direct dynamics) which is far

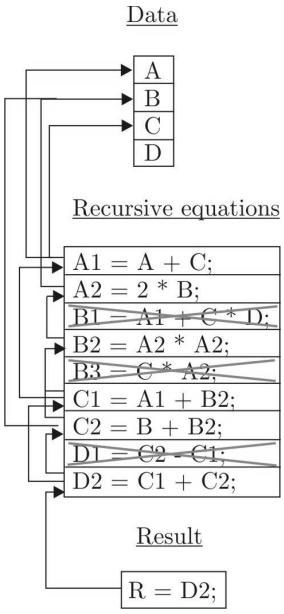

Fig. 5. Recusive elimination of superfluous equations

from being negligible. In terms of symbolic computation performances, the multiple scanning of the list (required by step 2 in the process) can be quite time consuming for large systems because string comparison is performed thousands of times to find and mark interdependencies. The use of supplementary flags in the list elements, not detailed here, allows us to find and mark the useful equations in an immediate manner. This allows us to reduce the elimination process to a few seconds, even for more than 200 d.o.f. multibody systems.

3.3. Recursive symbolic differentiation

As mentioned in the Section 2, sensitivity analysis of large constrained MBS led us to develop a specific procedure to symbollically differentiate a given recursive scheme with respect to a given (set of) parameter(s) p. In the context of differentiation, the equations produced by recursive multibody formalisms can be advantageously considered as interwoven functions ($f(g(h(...(x))))$). However, if the corresponding differentiation rules are applied blindly to a recursive scheme, we have observed that they produce a very large non-optimized symbolic output (ex. $\frac{d\ddot{u}}{\partial p}$) even for medium-sized multibody models: the interest of the recursive computation (see eqs. (13) and Fig. 4) is thus *completely* lost. Therefore, we take advantage of the elimination procedure described in the previous Section to solve this prob-

lem. When evaluating a given recursive scheme, we assume –apriori– that each equation depends, explicitly or not, on the set of system parameters or variables (ex.: $p_1, p_2, ...p_k$) with respect to which the differentiation must be performed. For instance in the following equation:

$$\text{AUXJ} = \text{AUXI} + 2 * \text{P} \tag{15}$$

AUXJ explicitly depends on the variable P via the second term. A priori, it may also depend implicitly on P via the first term AUXI. We thus systematically create and evaluate a new *recursive* variable, for instance AUXJ_P, for the partial derivative of the current equation with respect to P

$$\text{AUXJ\_P} = \text{AUXI\_P} + 2$$

even if, in the end, it appears that this new auxiliary variable is 0 or simply useless. If it is useless, the elimination process described previously will detect it and remove the corresponding equation from the list, before printing. Such a technique gives rise to a *compact recursive computation* of the required derivatives. To illustrate this, let us consider in (16) and (17), the symbolic evaluation of one element ($J_{(3,1)}$) of the Jacobian matrix $\frac{dx}{dq}$ of a position vector $x(q)$ associated with a kinematic chain composed of nine joints. In (16), the classical differentiation rule applied to $x(q)$, i.e. the resulting equation ($J_{(3,1)}$ = ...)**,** is far more consuming in terms of operators (220 $\{*, +, -\}$) than when obtained via the proposed recursive differentiation as (62 $\{*, +, -\}$).

```
J(3,1)  =  q8*(C1*C7*(-C3*S4*C5+S3*S5)-S1*(S2*(S3*S4*(-C5*C7+S5*C6*S7)
           +S7*(-C3*C5*C6+S3*C4*S6))+C7*(C2*C4*C5-S2*C3*S5))
           +S7*(C1*(C3*C4*S6+C6*(C3*S4*S5+S3*C5))-S1*C2*(-C4*S5*C6+S4*S6)))
           +D13*S1*S2+D14*(C1*S3+S1*S2*C3)+D15*(C1*S3+S1*S2*C3)
           +D16*(C1*(C3*S4*S5+S3*C5)+S1*(C2*C4*S5-S2*(-C3*C5+S3*S4*S5)))
           +D17*(C1*C6*(C3*S4*S5+S3*C5)+S1*(-S2*S3*(C4*S6+S4*S5*C6)
           +C6*(C2*C4*S5+S2* C3*C5))+S6*(C1*C3*C4-S1*C2*S4))
           +D18*(C1*S7*(C3*S4*C5-S3*S5)-S1*(S2*(S3*S4*(C5*S7+S5*C6*C7)
           +C7*(-C3*C5*C6+S3*C4*S6))+S7*(-C2*C4*C5+S2*C3*S5))
           +C7*(C1*(C3*C4*S6+C6*(C3*S4*S5+S3*C5))-S1*C2*(-C4*S5*C6+S4*S6)))
           +D19*(C1*S7*(C3*S4*C5-S3*S5)-S1*(S2*(S3*S4*(C5*S7+S5*C6*C7)
           +C7*(-C3*C5*C6+S3*C4*S6))+S7*(-C2*C4*C5+S2*C3*S5))
           +C7*(C1*(C3*C4*S6+C6*(C3*S4*S5+S3*C5))
           -S1*C2*(-C4*S5*C6+ S4*S6)));                                  (16)
```

```
RO22 = S1*S2; RO32 = -C1*S2;  RO82 = -S1*C2; RO92 = C1*C2;
RO23 = RO22*C3+C1*S3; RO33 = RO32*C3+S1*S3; RO53 = -RO22*S3+C1*C3;
RO63 = -RO32*S3+S1*C3; RO54 = RO53*C4+RO82*S4; RO64 = RO63*C4+RO92*S4;
RO84 = -RO53*S4+RO82*C4; RO94 = -RO63*S4+RO92*C4; RO25 = RO23*C5-RO84*S5;
RO35 = RO33*C5-RO94*S5; RO85 = RO23*S5+RO84*C5; RO26 = RO25*C6+RO54*S6;
RO36 = RO35*C6+RO64*S6; RO27 = RO26*C7-RO85*S7; RO87 = RO26*S7+RO85*C7;
RL23 =RO22*D13 ;  RL24 = RO23*D14;  JT34_1  =  RL23+RL24;
RL25 = RO23*D15 ; JT35_1  =  JT34_1+RL25;  RL26 = RO25*D16;
JT36_1 = JT35_1+RL26;  RL27 = RO26*D17;  RL28 = RO27*D18+RO87*q(8);
JT37_1 = JT36_1+RL27; JT38_1 = JT37_1+RL28; RL29 = RO27*D19;
J(3,1) = JT38_1+RL29;                                                    (17)
```

For larger models and in particular for the explicit direct dynamics (8) of constrained multibody systems, the advantage of the recursive differentiation is really significant. In fact, the *explosive* increase in the size of the classical differentiation technique is quite understandable since it amounts to destroying the recursivity of the original scheme, leading to an *in extenso* formulation like in example 16. Although the proposed recursive differentiation process is very consuming in terms of both memory storage and symbolic CPU time – because thousands of "potential" partial derivatives are computed –, these drawbacks are negligible in ROBOTRAN since the storage requirement is drastically controlled during the symbolic process [7].

4. Sensitivity analysis: illustrative results

A typical sensitivity analysis we propose to carry out is briefly illustrated here below. The multibody model is based on the one previously shown in Fig. (2), but involving semi-active suspensions. The force-velocity damper relationship is function of an electro-valve current i: the valve section, when reduced or enlarged, induces a higher or a lower damping coefficient of the suspension.

Two sensitivity computations are proposed here:

1. The first one concerns passenger comfort and relates to the sensitivity of the rms vertical acceleration with respect to the electro-valve current for a given road excitation;
2. The second one deals with car handling and aims at showing the influence of the front anti-roll bar torsion stiffness (thus $p = K_{arb}$) on the under/oversteering character of the car, for a given pilot steering input.

4.1. Comfort-oriented sensitivity

According to Fig. 2, the car is shaked on the 4-poster test bench (left/right out-of-phase noisy sinusoidal input (1 Hz, Amplitude : $\pm$ 10mm). The acceleration of the carbody center of mass** is given in Fig. 6$_{left}$ for illustrative purpose only. The rms value of the vertical acceleration can be computed

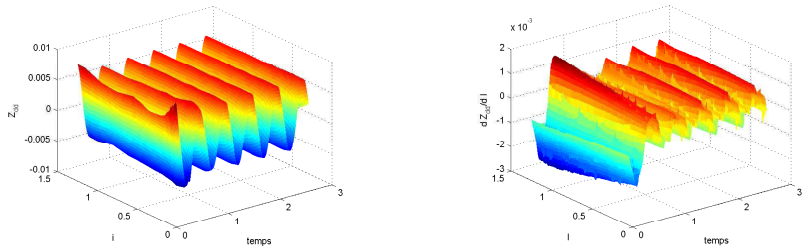

Fig. 6. Audi A6 on poster - Left: carbody acceleration, Right: carbody acceleration sensitivity

straightforwardly:

$$\ddot{z}^{rms} \triangleq \sqrt{\frac{1}{t_f - t_i}\int_{t_i}^{t_f} [\ddot{z}(t)]^2 \, dt} \quad = xxx\,\frac{m}{s^2} \quad \text{for the present simulation case.}$$

The sensitivity time history of the vertical acceleration $\ddot{z}_i$ for distinct damper electro-valve currents i (thus $p = i$) is shown in Fig. 6$_{right}$. While being more "cosmetic than pragmatic", this 3D plot contains relevant information for the manufacturers.

First of all, with respect to time t, the envisaged future trajectories will be composed of successive sections of different natures: smooth or uneven surfaces, sections with potholes or transverse discontinuities, etc. which will define successive time intervals for which dedicated objective function could be evaluated (like rms values or mean dissipated power, etc...).

With respect to the parameter p itself, it is useful to observe the evolution of the sensitivity with respect to some parameter range in order to make sure of the "robustness" of the dynamic behavior with respect to some parameter inaccuracy. Figure 7$_{left}$ shows, for instance the way the sensitivity of the rms acceleration $\ddot{z}_i^{rms}$ changes with respect to the current i (range: 0.3 A to 1.5 A). Since it is rather difficult to "feel" the meaning of the absolute value of $\ddot{z}_i^{rms}$, the latter will be advantageously translated in some pragmatical "metric", as for instance the percentage of output variation for a given relative (ex. 1 %) or absolute (ex. 0.1A) variation of the considered parameter.

** Other sensor points could easily be used, in particular considering the real passenger location.

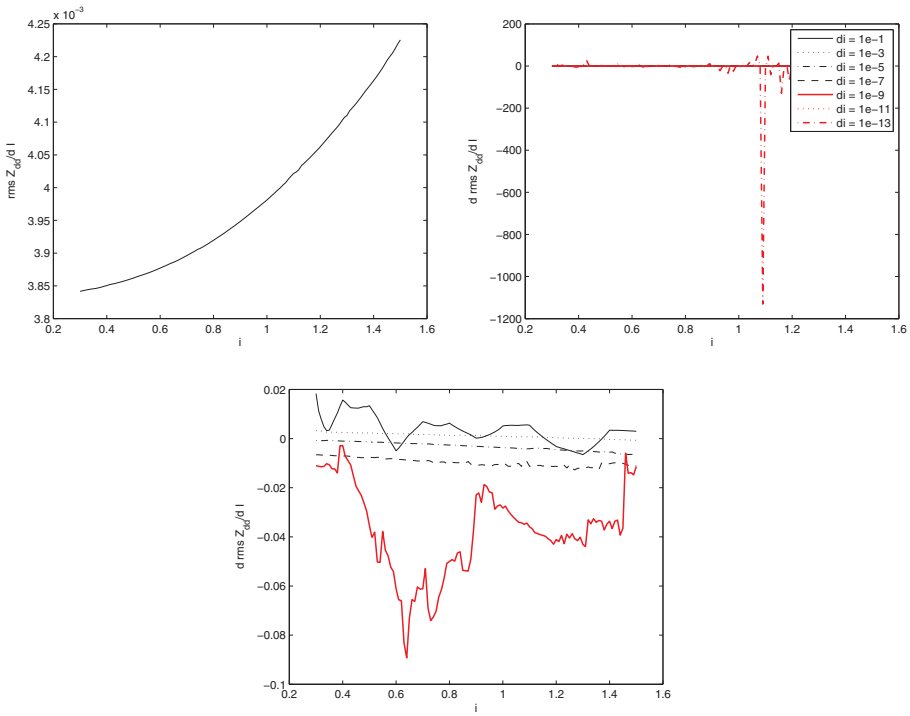

Fig. 7. Carbody rms acceleration – Left: influence of the electro-valve current, Right: repercussion of the numerical differentiation accuracy, down zoom of the middel figure

The purpose of Fig. $7_{right\ and\ down}$ is completely different: it simply illustrates the interest in using symbolical (i.e. analytical) differentiation technique, instead of numerical differentiation whose accuracy unavoidably depends on the parameter increment Δp. Moreover, with respect to symbolic differentiation, numerical approximation of gradients is unavoidably more time consuming since it requires at least two function evaluations: this additional cost is not negligible regarding the envisaged applications.

4.2. Handling-oriented sensitivity

In terms of car handling, lots of investigations can be carried out in terms of sensitivity with respect to suspension design. Within the frame of this paper, let us focus on the under/over-steering character of the car when negotiating a curve. In the literature, one can find various way to quantify this phenomenon, by comparing the so-called wheel slip angles of front and rear wheels, by considering or not the instantaneous centripetal acceleration of the car, etc. A simplest way – sufficient to illustrate our approach – consists in using for the objective function a more global parameter: the carbody slip

angle β illustrated in Fig. 8_{left}. β is the angle between the car longitudinal orientation $\vec{\mathbf{X}}$ and the car instantaneous center of mass velocity $\vec{\mathbf{V}}$. the case illustrated refers to an oversteering situation since the car turns "too much" with respect to the normal trajectory T.

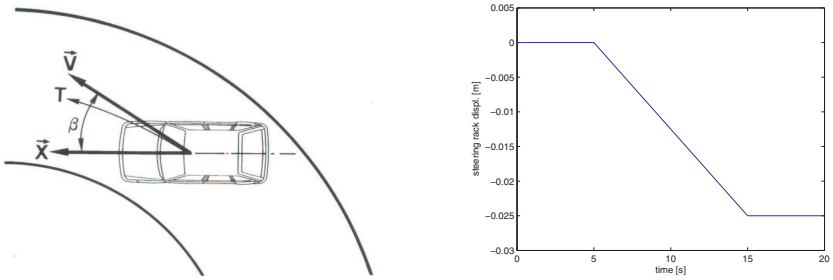

Fig. 8. Cornering situation - Left: car slip angle β, Right: steering rack displacement (imposed)

Figure 8_{right} represents the pilot steering input time history which is introduced via the steering rack displacement in the model.

As regards the over/understeering behavior of a vehicle negotiating a curve, a well-established observation is the following: increasing the roll stiffness K of the front [resp. rear] suspension, via an anti-roll bar for instance, increases [resp. decreases] the car understeering behavior. It is for instance possible to transform an understeering car (as we are daily using) into an oversteering one, simply by adding an anti-roll bar to the rear suspension. Without entering into details, the reason of this phenomenon comes from the nonlinear relationship between the lateral cornering force and the normal force: the global lateral – or centripetal – force on a given suspension (featuring left + right wheels) decreases when the vertical load transfer increases, which is the case when a stiffer anti-roll bar is used for that suspension.

For the proposed steering excitation, Fig. 9_{left} shows the Audi A6 trajectory for different values of the front suspension anti-roll bar torsions stiffness K. Fig. 9_{right} and 9_{down} relates to the vehicle configuration in steady state curving (line 's-s' in Figs. 8_{right} and 9_{left}). At this point, the vehicle – if stable – reaches a given slip angle β corresponding to a given amount of under/oversteer as shown in Fig. 9_{right}. Its sensitivity β_K with respect to the front anti-roll bar stiffness K is plotted in Fig. 9_{down}. On can observe a rather smooth evolution, which will be not necessarily the case for situations in which the lateral forces are for instance closer to their saturation point.

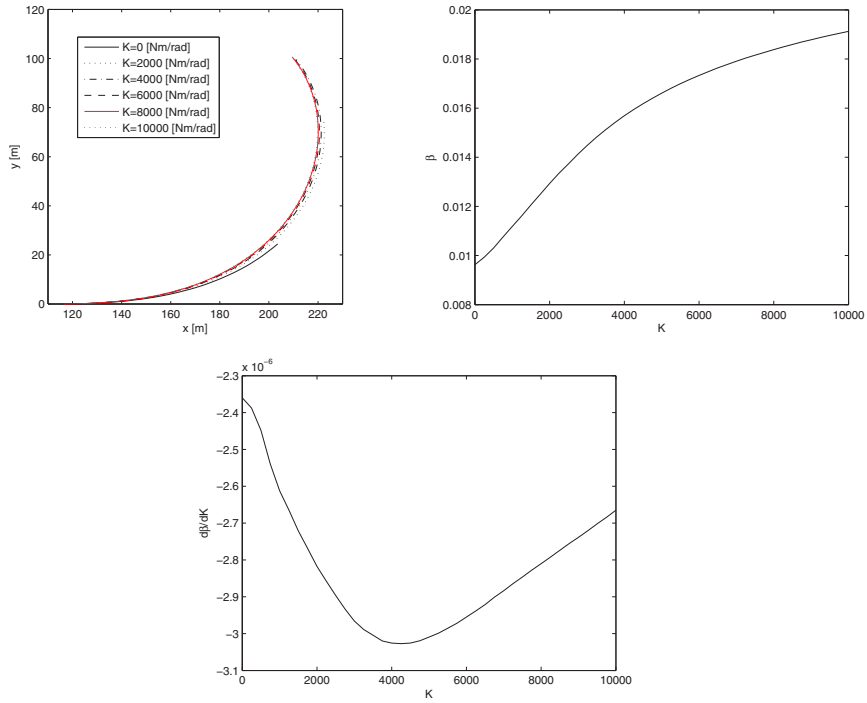

Fig. 9. Vehicle cornering behavior – Left: car trajectory, Right: car slip angle β versus K, Down: car slip angle sensitivity β_K versus K

5. Conclusions

The present work deals with the sensitivity analysis of constrained multibody models, the main applications being road vehicles equipped with modern suspensions. Considering the size of those systems and the necessity to use a rather refined multibody model to properly capture the required dynamic behaviors, the symbolic approach underlying the ROBOTRAN program is appealing for two reasons: the explicit and recursive computation of the generalized accelerations (direct dynamics) which are necessary for computing objective functions based on time integrations of the system (ex: car following a trajectory, ...) and the explicit – and also recursive – generation of the partial derivatives of the direct dynamics, with respect to a given (set of) parameter(s). All the process is based on the Coordinate Partitioning approach whose well-known computational cost is strongly reduced thanks to the symbolic simplification capabilities of the ROBOTRANprogram.
Direct method is used to compute and time integrate the multibody equations of motion simultaneously with the sensitivity differential equations, the main drawback with respect to other approaches (ex. the adjoint method) being the

numerical cost. Once again, the symbolic generation allows us to minimize this problem.
Additional applications – in the field of vehicle dynamics – will be carried out in the next future to compare and quantify the process performances with respect to the adjoint approach.
In terms of results interpretation, both the objective functions and the parameters can be of different nature (acceleration, force, power, etc.. for the former; dimension, current, stiffness, etc. for the latter). Thus, it is proposed to establish some *sensitivity metric* with the damper manufacturer, to decide what is "acceptable or not" for a given vehicle dynamic situation.

Acknowledgements

This research work was carried out in collaboration with the Belgian Company Tenneco Automotive (Sint-Truiden, Belgium): their assistance is gratefully acknowledged.

Manuscript received by Editorial Board, December 16, 2009;
final version, March 05, 2010.

REFERENCES

[1] R.A. Wehage and E.J. Haug, Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems. Journal of Mechanical Design 104, pp. 247-255, 1982.
[2] P. Fisette, T. Postiau, L. Sass and J.C. Samin, Fully Symbolic Generation of Complex Multibody Models, Mechanics Based Design of Structures and Machines, Marcel Dekker Inc., New York, 30, 1, pp. 1-32, 2002.
[3] J.Y. Ding, Z.K. Pan and L.Q. Chen, Second order adjoint sensitivity analysis of multibody systems described by differential-algebraic equations. Multibody System Dynamics 18, pp. 599-617, 2007.
[4] P. Eberhard, Analysis and Optimization of Complex Multibody Systems using Advanced Sensitivity Analysis Methods. ZAMM, 76, 3, pp. 40-43, 1996.
[5] H. B. Pacejka, Tyre and Vehicle Dynamics (Second Edition). Butterworth-Heinemann (Elsevier), Amsterdam, 2006.
[6] S. Duym, Simulation, Modeling and Identification, for an Automotive Shock Absorber in the Context of Vehicle Dynamics. Vehicle System Dynamics 33, 261-185, 2000.
[7] J.C. Samin and P. Fisette, Symbolic Modeling of Multibody System. Kluwer Academic Publishers, Dordrecht, the Nederlands, 2003.
[8] M. Delannoy, P. Boon, B. Vandersmissen and P. Fisette, Modeling hybrid multibody systems: application to vehicle hydraulic semi-active suspensions. Eccomas – Multibody Dynamics 2005, Madrid, Spain, 21-24, June 2005.
[9] W.F. Milliken and D.L. Milliken, Race Car Vehicle Dynamics. Society of Automotive Engineers, Inc., Warrendale, USA, 1995.

Wykorzystanie symbolicznego generowania równań w analizie wrażliwości zawieszenia samochodowego na zmiany parametrów

S t r e s z c z e n i e

Cel przedstawionych badań jest związany z analizą wrażliwościową wpływu, jaki na komfort jazdy i właściwości jezdne pojazdu drogowego mają parametry technologiczne jego zawieszenia. Przewiduje się, że zawieszenie będzie miało charakter półaktywny, co implikuje podejście polegające na modelowaniu hybrydowym, w którym wykorzystuje się trójwymiarowy model całego samochodu – w konkretnym przypadku jest to Audi A6 – w połączeniu z elektro-hydraulicznym modelem amortyzatorów zawieszenia. Wrażliwość modelu bada się, by określić wartości parametrów – a nie w celach optymalizacyjnych – gdyż znajomość tych wielkości jest ogromnie interesująca dla producentów amortyzatorów.

Ważną kwestią w tych badaniach jest przyjęcie funkcji celu, które oparte są na całkowaniu pewnych wielkości podczas całego czasu realizacji zadanej trajektorii. Celem może być – na przykład – określenie wrażliwości średniokwadratowego przyspieszenia nadwozia (komfort) lub charakterystyki przewracania się samochodu (własności trakcyjne) na zmiany parametrów zawieszenia. Z jednej strony, dokładność obliczeń różnych pochodnych cząstkowych może być znakomicie zwiększona dzięki możliwościom programu do symulacji układów wieloczłonowych ROBOTRAN, który wykorzystuje postać symboliczną równań. Z drugiej strony, korzystny dla wydajność procesu obliczeniowego jest również fakt, że równania ruchu układu wieloczłonowego są sformułowane w postaci rekursywnej. W przypadku tak zwanej metody bezpośredniej, która leży u podstaw analizy wrażliwości, równania te muszą być całkowane w dziedzinie czasu względem zarówno współrzędnych uogólnionych jak i ich pochodnych cząstkowych.