COMPUTATIONAL MECHANICS 2004
20th conference with international participation
Hrad Nečtiny
November 8 - 10, 2004

RESOLUTION OF CONSTRAINT EQUATIONS FOR REAL-TIME MODEL OF VEHICLE SUSPENSION

Jakub TOBOLÁŘ[1], **Wolfgang RULKA**[2]

Abstract: *This paper presents one of the approaches for the reduction of suspension models containing kinematic closed loops for real-time simulation. The model reduction is based on the ODE formulation of the equations of motion in the independent coordinates together with subsystem-local solution of constraint equations describing kinematic loops. Furthermore, different methods for formulating the constraint equations are mentioned. The implementation of the reduction technique in the multibody simulation package SIMPACK and some simulation results are presented as well.*

Key words: *Model reduction, real-time, suspension*

1. INTRODUCTION

The simulation of multibody systems (MBS) has become a standard method both for off-line and on-line analyses of vehicles. For the latter real-time simulation models are necessary that are nowadays mostly generated on the basis of generic models by special simulation programs (e. g. ve–DYNA [1]). Since the extension of such generic models is restricted, the general MBS simulation tools must be used to generate advanced vehicle models. But to perform the time integration of such models in real-time the model reduction has to be performed based on the expert know-how of the design engineer. Since the vehicle suspension systems generally imply kinematic closed loops, the reduction techniques have been developed for general MBS simulation tools based on the recursive formalism. These techniques simplify the model reduction process and avoid kinematic loops in the suspension model.

2. SIMPLIFICATION OF SUSPENSION MODELS

A typical vehicle suspension (Fig.1) consists of the wheel carrier 3, rods 2, 4, 5, 6 and spring and damper 9, 10. To simplify the full suspension models containing kinematic closed loops that generally lead to the solution of differential algebraic equations (DAE), several approaches have been developed, see [1, 2, 3, 4, 5]. Due to the model reduction, the algebraic equations describing the kinematic loops are avoided and equations of motion of the vehicle form ordinary differential equations (ODE) that can be solved efficiently. Moreover, assuming that rod dynamics is negligible for the overall behaviour of the vehicle, the mass and inertia of these rod bodies may not be taken into account in a suspension submodel. Therefore a higher evaluation efficiency can be reached due to the reduced number of equations of motion. The idea of one of the model reduction techniques – the so-called macro-joint [5] – and new advances will be described below.

[1]Ing. Jakub Tobolář, Institute of Robotics and Mechatronics, German Aerospace Center (DLR), P.O.Box 1116, D-82230 Wessling, Germany, jakub.tobolar@dlr.de
[2]Dr.-Ing. Wolfgang Rulka, INTEC GmbH, Argelsrieder Feld 13, D-82234 Wessling, Germany, wolfgang.rulka@simpack.de

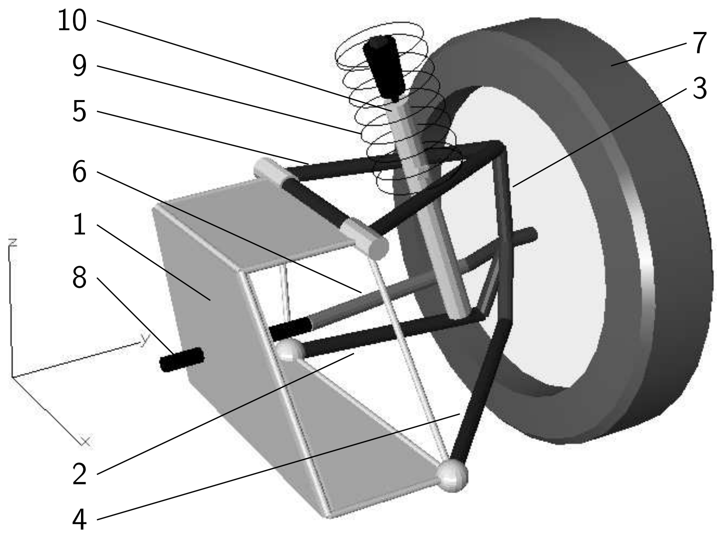

Fig. 1. Structure of typical independent suspension

3. MACRO-JOINT

Generally the equations of motion of an MBS are found as

$$\boldsymbol{M}(\boldsymbol{p},t)\ddot{\boldsymbol{p}} = \boldsymbol{h}(\boldsymbol{p},\dot{\boldsymbol{p}},t) + \boldsymbol{\Phi}_p^T(\boldsymbol{p},t)\boldsymbol{\lambda}, \quad (1)$$

$$\boldsymbol{0} = \boldsymbol{\phi}(\boldsymbol{p},t), \quad (2)$$

see e. g. [6]. Here $\boldsymbol{M}$ is the inertia matrix and vector $\boldsymbol{h}$ summarises the applied and Coriolis forces. The last term in (1) represents reaction forces. For redundant coordinates summarised in a vector $\boldsymbol{p}$ the implicitly given equations (2) of holonomic constraints are defined. The number of Lagrangian multipliers $\boldsymbol{\lambda}$ is equal to the number of constraints (2). The equations (1) interrelate with constraints (2) by the distribution matrix $\boldsymbol{\Phi}_p = \partial\boldsymbol{\phi}/\partial\boldsymbol{p}^T$.

For a recursive formulation of the equations of motion which is the basis of the presented reduction method, the number of coordinates $\boldsymbol{p}$ is equal to the number of freedom of corresponding joints of mechanical systems with kinematic chain or tree topology. For constrained mechanical systems with closed kinematic loops, each loop contains a "cut-joint" that is described by the constraints (2). In this case the vector $\boldsymbol{\lambda}$ summarises generalised constraint forces of these cut-joints. The number n_{z_u} of independent coordinates $\boldsymbol{z}_u$ corresponds with the number n_f of degrees of freedom of such a constrained mechanical system and is smaller than the number n_p of coordinates $\boldsymbol{p}$: $n_{z_u} < n_p$.

Introducing matrix $\boldsymbol{R}$ defined by

$$\dot{\boldsymbol{p}} = \boldsymbol{R}\dot{\boldsymbol{z}}_u \quad (3)$$

the generalised constraint forces $\boldsymbol{\lambda}$ can be eliminated from (1) and at the same time the number of equations of motion can be reduced [2]. The equations of motion result in

$$\hat{\boldsymbol{M}}\ddot{\boldsymbol{z}}_u = \hat{\boldsymbol{h}}, \quad (4)$$

with $\hat{\boldsymbol{M}} = \boldsymbol{R}^T\boldsymbol{M}\boldsymbol{R}$, $\hat{\boldsymbol{h}} = \boldsymbol{R}^T(\boldsymbol{h} + \boldsymbol{M}\dot{\boldsymbol{R}}\dot{\boldsymbol{z}}_u)$ whereby the time derivative of (3) has been introduced. Generally special MBS-formalisms are used to define the independent coordinates $\boldsymbol{z}_u$, to build the matrix $\boldsymbol{R}$ and to generate appropriate equations of motion (4). For the recursive formalism the following idea on how to overcome the calculation of $\boldsymbol{R}$ has been realised.

Assuming that the mass and inertia of suspension rods (Fig.1) can be neglected, the rods are modelled as massless links kinematically defined by a so-called *characteristic joint couple* [7]. Then the matrix $\boldsymbol{R}$ reads for the vehicle model (compare (3)):

$$\begin{bmatrix} \dot{\boldsymbol{p}}_1 \\ \dot{\boldsymbol{p}}_2 \\ \vdots \\ \dot{\boldsymbol{p}}_i \\ \vdots \\ \dot{\boldsymbol{p}}_{n_p} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ & \boldsymbol{E} & \dots & \boldsymbol{0} & \dots & \boldsymbol{0} \\ & & \ddots & \vdots & & \vdots \\ & & & \boldsymbol{R}_i & \dots & \boldsymbol{0} \\ & sym. & & & \ddots & \vdots \\ & & & & & \boldsymbol{E} \end{bmatrix}}_{\boldsymbol{R}} \begin{bmatrix} \dot{\boldsymbol{z}}_{u1} \\ \dot{\boldsymbol{z}}_{u2} \\ \vdots \\ \dot{\boldsymbol{z}}_{ui} \\ \vdots \\ \dot{\boldsymbol{z}}_{un_f} \end{bmatrix}, \tag{5}$$

with identity matrix $\boldsymbol{E}$. Since the Matrix $\boldsymbol{R}_i$ defines the dependency of the joint coordinates $\boldsymbol{p}_i$ of the wheel carrier i on the independent coordinates $\dot{\boldsymbol{z}}_{ui}$ of the same body and it is the only matrix that must be specified, it can be calculated already during the evaluation of kinematics of the multibody model.

3.1. Kinematics of the wheel carrier joint

The vector $\boldsymbol{r}_i(\boldsymbol{p}_i)$ of relative translation and the tensor $\boldsymbol{A}_i(\boldsymbol{p}_i)$ of relative rotation define relative kinematics of the wheel carrier joint, i. e. the relative position and orientation of the wheel carrier to the chassis. The relative translational kinematics of the carrier joint is defined as

$$\boldsymbol{r}_i = \boldsymbol{r}_i(\boldsymbol{p}_i), \tag{6}$$
$$\boldsymbol{v}_i = \boldsymbol{J}_{Ti}\dot{\boldsymbol{p}}_i + \overline{\boldsymbol{v}}_i, \tag{7}$$
$$\boldsymbol{a}_i = \boldsymbol{J}_{Ti}\ddot{\boldsymbol{p}}_i + \overline{\boldsymbol{a}}_i, \tag{8}$$

with the Jacobian matrix of translation $\boldsymbol{J}_{Ti}$. The relative rotational kinematics can be derived similarly.

Due to the *partitioning* of the joint coordinates $\boldsymbol{p}_i$ (see e. g. [2]) of wheel carrier i into independent one $\boldsymbol{z}_{ui}$ and dependent one $\boldsymbol{z}_{ai}$ (i. e. $\boldsymbol{p}_i = [\boldsymbol{z}_{ai}^T \ \boldsymbol{z}_{ui}^T]^T$) the following constraint equations (and their time derivatives) of simplified suspension subsystem are obtained:

$$\boldsymbol{\phi}(\boldsymbol{z}_{ai}, \boldsymbol{z}_{ui}) = \boldsymbol{0}, \tag{9}$$
$$\boldsymbol{\Phi}_{z_{ai}}\dot{\boldsymbol{z}}_{ai} + \boldsymbol{\Phi}_{z_{ui}}\dot{\boldsymbol{z}}_{ui} = \boldsymbol{0}, \tag{10}$$
$$\boldsymbol{\Phi}_{z_{ai}}\ddot{\boldsymbol{z}}_{ai} + \boldsymbol{\Phi}_{z_{ui}}\ddot{\boldsymbol{z}}_{ui} + \dot{\boldsymbol{\Phi}}_{z_{ai}}\dot{\boldsymbol{z}}_{ai} + \dot{\boldsymbol{\Phi}}_{z_{ui}}\dot{\boldsymbol{z}}_{ui} = \boldsymbol{0}, \tag{11}$$

whereby only coordinates $\boldsymbol{z}_{ai}$ and $\boldsymbol{z}_{ui}$ of the wheel carrier are involved. In equations (9), (10) and (11) the characteristic joint couple of massless links are involved.

To calculate wheel carrier kinematics both $\boldsymbol{z}_{ai}$ and $\boldsymbol{z}_{ui}$ have to be defined. Generally these coordinates are included in vector $\boldsymbol{p}$ of the overall multibody model, but since the dependent coordinates $\boldsymbol{z}_{ai}$ are used only locally for the simplified suspension subsystem, the constraint equations (9) to (11) can be involved directly in equations (6) to (8) of joint kinematics as follows:

$$\boldsymbol{r}_i = \boldsymbol{r}_i(\boldsymbol{z}_{ai}, \boldsymbol{z}_{ui}), \tag{12}$$
$$\boldsymbol{v}_i = \boldsymbol{J}_{Ti}\boldsymbol{R}_i\dot{\boldsymbol{z}}_{ui} + \overline{\boldsymbol{v}}_i = \hat{\boldsymbol{J}}_{Ti}\dot{\boldsymbol{z}}_{ui} + \overline{\boldsymbol{v}}_i, \tag{13}$$
$$\boldsymbol{a}_i = \boldsymbol{J}_{Ti}\boldsymbol{R}_i\ddot{\boldsymbol{z}}_{ui} + \boldsymbol{J}_{Ti}\dot{\boldsymbol{R}}_i\dot{\boldsymbol{z}}_{ui} + \overline{\boldsymbol{a}}_i = \hat{\boldsymbol{J}}_{Ti}\ddot{\boldsymbol{z}}_{ui} + \hat{\boldsymbol{a}}_i, \tag{14}$$

and similarly for rotational kinematics. Dependent coordinates $\boldsymbol{z}_{ai}$ are then defined and used only locally by relative kinematics of the wheel carrier joint. The joint described by equations (6) to (8) is called *macro-joint* since it reflects the kinematics of the whole (suspension) subsystem. Due to the elimination of coordinates $\boldsymbol{z}_{ai}$ from subvector $\boldsymbol{p}_i$ (i. e. $\boldsymbol{p}_i := \boldsymbol{z}_{ui}$) the equations of motion of the overall multibody model result in ODE

$$\boldsymbol{M}(\boldsymbol{p}, t)\ddot{\boldsymbol{p}} = \boldsymbol{h}(\boldsymbol{p}, \dot{\boldsymbol{p}}, t) \tag{15}$$

that can be solved efficiently during real-time simulation.

3.2. Solution of nonlinear constraint equations

To evaluate the kinematics of the macro-joint the coordinates $\boldsymbol{z}_{ai}$, $\dot{\boldsymbol{z}}_{ai}$ and $\ddot{\boldsymbol{z}}_{ai}$ must be specified. To obtain the derivatives $\dot{\boldsymbol{z}}_{ai}$ and $\ddot{\boldsymbol{z}}_{ai}$ just linear equations (10) and (11) must be solved, for $\boldsymbol{z}_{ai}$ the solution of generally highly nonlinear constraints (9) is necessary. To solve such nonlinear algebraic equations, common iterative numerical methods can be used [8].

To avoid an iterative solution that consumes an important amount of computing time and is unsuitable for real-time simulation (because of the number of iteration steps that cannot be predicted), the integration of velocity equations

$$\dot{\boldsymbol{z}}_{ai} = -\boldsymbol{\Phi}_{z_{ai}}^{-1}\boldsymbol{\Phi}_{z_{ui}}\dot{\boldsymbol{z}}_{ui} \tag{16}$$

has been suggested to solve the position problem [2]. But for long simulations the position constraints (9) can be violated due to round-off errors of numerical integration since they are not involved in the time integration process. Therefore the stabilisation techniques have been suggested to overcome instability of numerical solution. From the range of stabilisation techniques that are applied during the numerical integration of the model, the following two are applicable for the macro-joint as well. The first one is Baumgarte stabilisation [9] that replaces equation (16) by

$$\dot{\boldsymbol{z}}_{ai} = -\boldsymbol{\Phi}_{z_{ai}}^{-1}(\boldsymbol{\Phi}_{z_{ui}}\dot{\boldsymbol{z}}_{ui} + \alpha_B\boldsymbol{\phi}(\boldsymbol{z}_{ai}, \boldsymbol{z}_{ui})) \tag{17}$$

thus involving position constraints (9). The constant α_B must be chosen appropriately to guarantee the stability of the numerical solution. Since the suspension is a relatively small subsystem, the choice of α_B value is generally harmless and it has been proven that $\alpha_B = 1$ is satisfactory.

The second possibility to stabilise the numerical solution of (16) is to use the projection method. Hereby the initial solution $\tilde{z}_{ai}$ given by numerical integration of (16) will be projected in a subspace of correct vectors $\boldsymbol{z}_{ai}$. The projection can be performed by the iterative Newton method:

$$\boldsymbol{z}_{ai}^{(k+1)} = \boldsymbol{z}_{ai}^{(k)} - \boldsymbol{\Phi}_{z_{ai}}^{-1}(\boldsymbol{z}_{ai}^{(k)})\boldsymbol{\phi}(\boldsymbol{z}_{ai}^{(k)}) \quad \text{with} \quad \boldsymbol{z}_{ai}^{(0)} = \tilde{\boldsymbol{z}}_{ai}. \tag{18}$$

To avoid a time consuming iterative process again, the number of iterations must be limited. Two steps ($k = 1$) at most are generally satisfactory to guarantee the accuracy of the macro-joint.

3.3. Formulation of constraint equations

In order to define local macro-joint constraints (9) a recursive formalism can be applied similar to that used for the equations of motion of the overall mechanical system [10]. Consequently, to define the constraints the relative coordinates are utilised by default. Using relative coordinates just the minimal number of constraints must be defined, however they are of high nonlinearity.

Since the dependent coordinates $\boldsymbol{z}_{ai}$ occur only macro-joint locally it is possible to formulate the constraint equations (9) (and consequently also the kinematics (12) to (14)) in any other coordinates. Therefore the formulation in so-called *natural coordinates* (see [2]) has alternatively been derived.

The natural coordinates are formed by Cartesian coordinates of selected points and unit vectors firmly linked with bodies. The points and vectors are defined on the joints and are shared by the bodies linked by the appropriate joint. The advantage of using natural coordinates for the macro-joint is that there is just one set of constraint equations defined for many types of vehicle suspensions based on the five-link suspension (Fig.2). Thus it is not necessary to involve the formalism compiling the constraints as it would be in the case of the relative coordinates. Moreover, the high nonlinearity of constraint equations is avoided because they are at most quadratic (for five-link suspension the constraints describe just the constant distance between the points fixed on bodies). Only in some special cases such as for McPherson suspension the new set of equations must be specified.

4. SIMULATION

The presented reduction method has been implemented and tested in the multibody simulation package SIMPACK, see [10]. The "quarter-car" model with five-link suspension (Fig.2) has been defined to compare different methods for formulating the algebraic equations. The suspension has been periodically excited in the tyre contact point with a quasi-stochastic signal that acts only in vertical direction. The suspension is modelled as a complete suspension in the reference model *5link_ref*, see Tab. 1. By reduced models the rods have been neglected and the suspension kinematics has been reproduced by reduction techniques presented in this paper (models *5link_macro* and *5link_natco_**). Besides these reduction methods, "Virtual Axle" reduction in model *5link_table* has also been used – a simple reduction technique that interpolates suspension kinematics by splines, see [4].

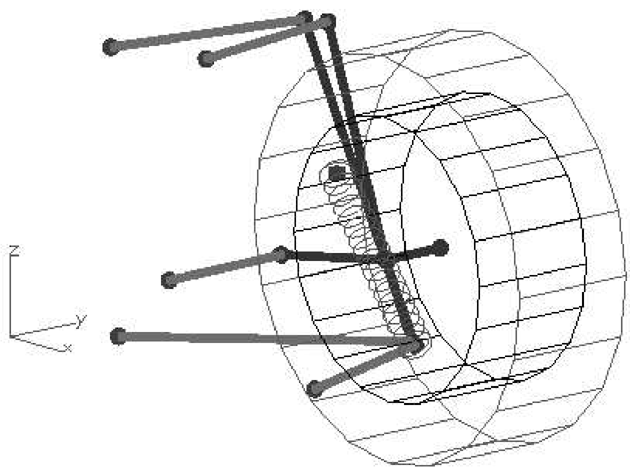

Fig. 2. Multibody model of five-link suspension

5. RESULTS

Altogether seven variants of five-link suspension have been modelled. For time integration the Intel Xeon (Pentium IV) processor has been used running at 3,2 GHz with 4 GB RAM and Microsoft Windows XP operating system. All reduced models have been analysed with

Tab. 1. Variants of simulated model

Model	Description
5link_ref	reference model including kinematic loops
*5link_natco_**	macro-joint in natural coordinates,
* := *iter*	iterative solution of $\boldsymbol{z}_{ai}$, $\boldsymbol{\Phi}_{z_{ai}}$ given numerically
* := *iter_a*	iterative solution of $\boldsymbol{z}_{ai}$, $\boldsymbol{\Phi}_{z_{ai}}$ given analytically
* := *proj*	time integrated solution $\tilde{\boldsymbol{z}}_{ai}$ stabilised by projection
* := *baumgte*	time integrated solution $\boldsymbol{z}_{ai}$ stabilised by Baumgarte
5link_macro	macro-joint in relative coordinates, Baumgarte stabilisation used
5link_table	spline interpolated kinematics (Virtual Axle)

explicit Euler integrator running with the sampling rate $h = 0.001$ s to compare the CPU time needed for time integration. Moreover, the SIMPACK integrator SODASRT (based on BDF numerical integration method, see [11]) with restricted variable time step has been considered for reference model *5link_ref* since this model leads to DAE and the Euler integrator is therefore not applicable.

As can be seen in Tab. 2, all simulations have been performed in real-time (speed-up factor of all models is $k_{EZ} = T_{CPU}/T_{sim} < 1$). The spline interpolated kinematics (model *5link_table*) has been proven to be the fastest method. The macro-joint approach is slightly slower than spline interpolated kinematics, whereby the formulation in natural coordinates (model *5link_natco_baumgte*) needs more computer time than that of relative coordinates (model *5link_macro*). The macro-joint with the iterative solution of $\boldsymbol{z}_{ai}$ proves to be the slowest method. Concerning the model accuracy, all the reduced models show comparable results, see Fig.3

6. CONCLUSIONS

The reduction method for suspension models has been presented. It is based on the formulation of equations of motion in independent coordinates and on the subsystem-local solution of algebraic equations describing kinematic loops. For the local solution of algebraic equations the iterative method has been proven to be less favourable due to the extra computing time consumption compared with the stabilised solution of time integrated (local) dependent

Tab. 2. Computing times T_{CPU} of simulated model, explicit Euler integrator, time of simulation $T_{sim} = 20$ s

Model	T_{CPU} [s]	k_{EZ} [-]
5link_ref	6,92*	0,34
5link_natco_iter	8,47	0,42
5link_natco_iter_a	6,41	0,32
5link_natco_proj	4,05	0,20
5link_natco_baumgte	4,22	0,21
5link_macro	3,23	0,16
5link_table	2,88	0,14

* SODASRT, max. time step $h_{max} = 0{,}001$s

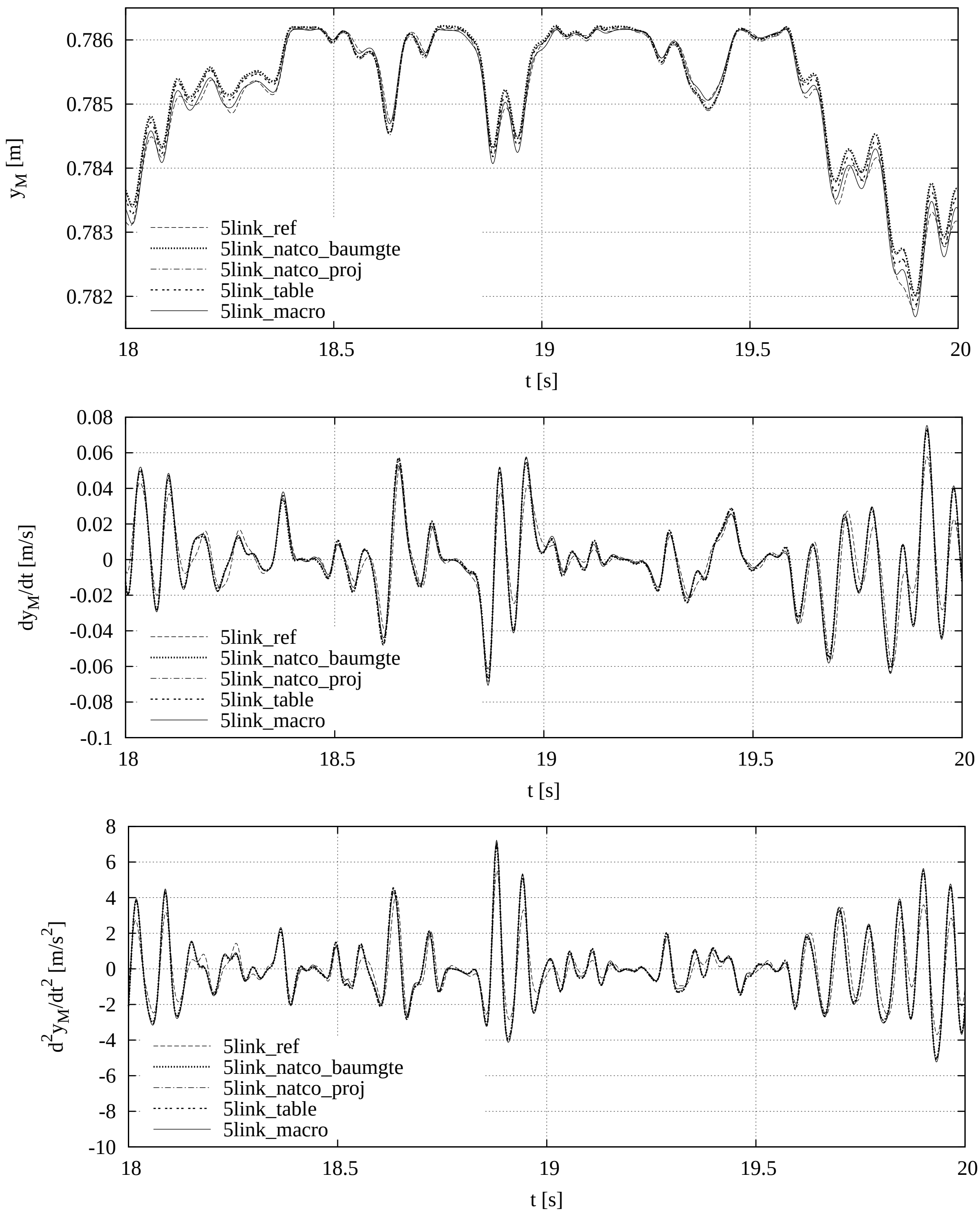

Fig. 3. Position, velocity and acceleration of the wheel centre point M in lateral direction

coordinates z_{ai}. The algebraic equations have been defined both in relative and natural coordinates. The first is preferred due to less computing time, the second one requires less input parameters (just absolute coordinates of linking points between rods and chassis or wheel carrier, respectively) thus being more user-friendly. The presented macro-joint needs slightly more computing time than spline interpolated kinematics (Virtual Axle, see [4]). Nevertheless, the transformation process from complete to reduced model is by the macro-joint based on the full model parametrisation, thus enabling the transformation to be performed much more safely and easily than by Virtual Axle. The simulation results in the time domain indicate just small deviations of reduced models compared to the complete reference model of suspension. Therefore the reduction method is suitable for the real-time simulation of vehicles.

REFERENCES

[1] Rill, G.: Simulation von Kraftfahrzeugen, Vieweg Verlag, 1994

[2] García de Jalón, J. and Bayo, E.: Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge, Springer-Verlag New York, 1994

[3] Choi, G. J. et al.: Vehicle Modeling Methods for Real-Time Dynamic Simulation Using Suspension Composite Joints, Mechanics of Structures and Machines, 2000, 28(4):303–321

[4] Tobolář, J.: The virtual axle – a simplified vehicle suspension model for real-time applications, In: Computational mechanics 2002, University of West Bohemia in Pilsen, Nečtiny, 2002, 461–468

[5] Rulka, W. and Pankiewicz, E.: MBS approach to generate equations of motions for HIL-simulations in vehicle dynamics, In: ECCOMAS Thematic Conference on Advances in Computational Multibody Dynamics, Lisbon, 2003

[6] Schiehlen, W.: Multibody System Dynamics: Roots and Perspectives, Multibody System Dynamics, 1997, 1(2):149–188

[7] Woernle, Ch.: Ein Verfahren zur behandlung von kinematischen Schleifen und zur Rückwärtstransformation bei Industrierobotern, ZAMM, 1987, 67:T 166–T 168

[8] Kelley, C.T.: Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995

[9] Baumgarte, J.: Stabilisation of constraints and integrals of motion in dynamical systems, Computer Methods in Applied Mechanics and Engineering, 1972, 1:1–16,

[10] Rulka, W.: Effiziente Simulation der Bewegungsgleichungen mechatronischer Systeme für industrielle Anwendungen, PhD thesis, TU-Wien, 1998

[11] Arnold, M.: Numerische Grundlagen von Simulationswerkzeugen der Fahrzeugsystemdynamik, CCG Seminar TV 1.03, Carl-Cranz-Gesellschaft, Oberpfaffenhofen, 2002