Simulation Based Design of Automotive Systems

Werner O. Schiehlen

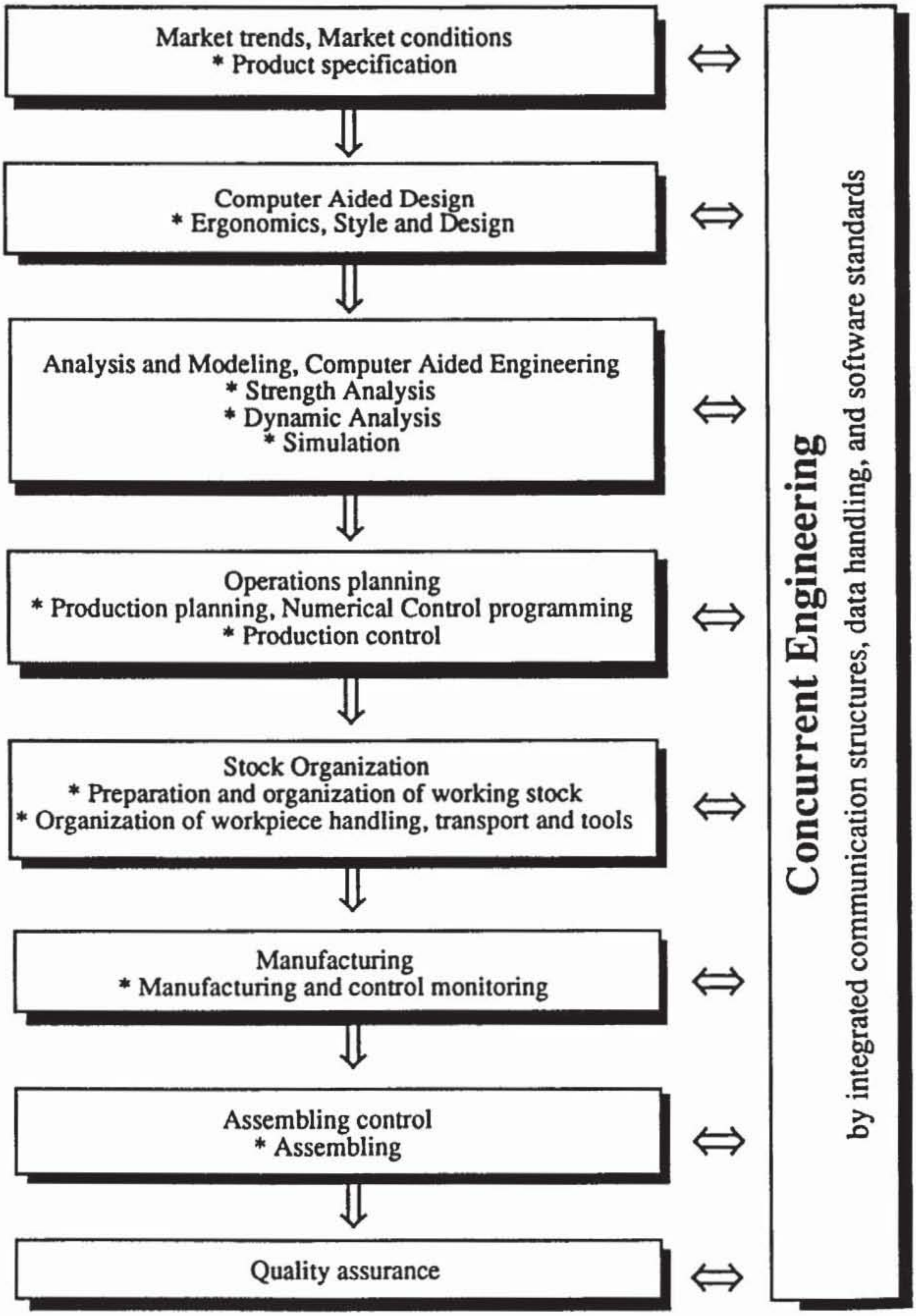
Institute B of Mechanics, University of Stuttgart, W-7000 Stuttgart 80, FRG

Abstract: The design of automotive systems using simulation tools features cost reduction and quality enhancement. This paper presents two basic approaches. The first approach deals with the application of CAD data bases to the evaluation of input data for multibody system formalisms, most adequate for automotive system modeling. An object oriented data model for multibody systems is presented. The second approach covers the development of an integrated simulation tool for automotive vehicles and the corresponding animation facilities. Driving comfort is related to the human perception of mechanical vibration. A companion paper deals with the optimization of automobile parameters using the multibody systems approach.

Keywords: CAD data base / object oriented data model / modeling / multibody systems / equations of motion / vehicle dynamics / simulation / animation / driving comfort

1 Introduction

The strong worldwide competition of automotive industries results in a large variety of automobiles to be developed in shorter and shorter periods. Thus, the classical method of automobile design via intensive experimental testing of prototypes is no longer economically feasible. Therefore, the dynamical behavior of a vehicle has to be simulated during the development process simultaneous with the overall design of the automobile. Most of the data for dynamical modeling and simulation are available in the CAD data base, blueprints and drawings may be omitted. This advanced method is part of Concurrent Engineering (CE) defined as an approach for designing and validating a product, its manufacturing process, and its quality control, all at the same time, see Figure 1. In particular, Concurrent Engineering is superior to the traditional sequential engineering with respect to the time required for the development of a new product. An integrated information processing results in an essential time saving, Figure 2.


Market trends, Market conditions
* Product specification
Computer Aided Design
* Ergonomics, Style and Design
Analysis and Modeling, Computer Aided Engineering
* Strength Analysis
* Dynamic Analysis
* Simulation
Operations planning
* Production planning, Numerical Control programming
* Production control
Stock Organization
* Preparation and organization of working stock
* Organization of workpiece handling, transport and tools
Manufacturing
* Manufacturing and control monitoring
Assembling control
* Assembling
Quality assurance
Concurrent Engineering
by integrated communication structures, data handling, and software standards

·Figure 1. Concurrent Engineering Approach

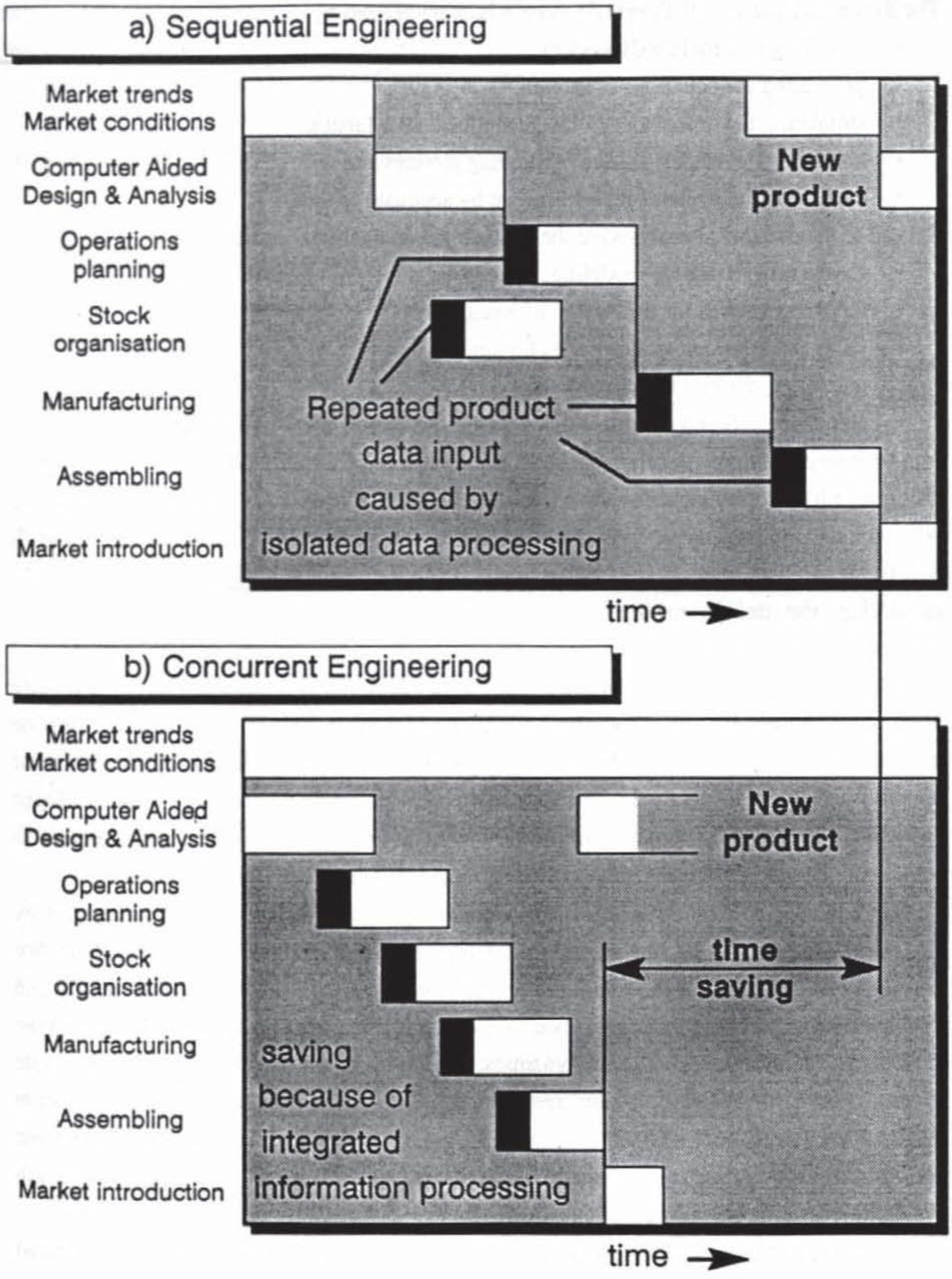


Figure 2. Sequential versus Concurrent Engineering

The dynamical analysis of an vehicle system is characterized by

- modeling as a multibody system,
- generating the equations of motion,
- simulating the trajectories of the generalized coordinates,
- animating the vehicle system by moving pictures and
- evaluating the dynamical performance by adequate criteria.

Related to the dynamical analysis are the following computational aspects:

- CAD software for the modeling,
- CAE formalisms for generation of equations,
- ODE and DAE integration codes for simulations,
- computer graphics for animations and
- signal analysis and optimization codes for evaluation of the performance.

In a first step, a unique description of all the elements of a multibody system is required. Using an object oriented software approach, the data of multibody systems elements are defined independently of the formalism applied for the generation of equations of motion. It will be shown how element data are extracted from the original CAD data of a vehicle. In particular, the design process and the dynamical analysis of the vehicle can be handled concurrently.

It has been proven that symbolically generated equations of motion are computationally more efficient than numerically derived equations of motion. This is valid not only for time integration during simulation but also for parameter variation during optimization or sensitivity analysis, respectively. The formalism NEWEUL will be presented in detail. Open and closed kinematical loops in multibody systems will be considered with respect to computational efficiency most important for simultaneous engineering applications.

Automobiles are highly nonlinear dynamical systems which may be investigated by numerical simulation or by linearization techniques resulting in eigenfrequency analysis. For the numerical simulation the available integration codes have to thoroughly tested and implemented in software packages. Usually more than one code is required to handle all the different problems in vehicle system dynamics. The choice of the computer code may be made by the user or the software system, respectively. Due to the nonlinearity, the simulation results show usually very irregular or chaotic motions. Then, signal analysis techniques from nonlinear dynamics have to be included in the investigation. This is also true for the strength evaluation requiring stochastic methods from material sciences.

Using the CAD data of automotive systems and the time histories of the generalized coordinates, the motion can be made visible by animation. This is a favorable approach for checking the simulation and to obtain a general idea of the motion. However, for an engineering improvement of a highly developed system like an automobile, special criteria for rating are necessary. Two of the most essential criteria are the riding comfort and the riding safety. The comfort is related to the human perception of mechanical vibrations while the safety can be rated via the vertical dynamic tire load. It will be shown how the first criterion may be introduced in the Concurrent Engineering process, too.

The final step, finding optimal parameters for the automotive system, will be treated in the companion paper by Dieter Bestle entitled 'Optimization of Automotive Systems.'

2 Multibody Systems Modeling

Road vehicles can be modeled properly as multibody systems for the design and the analysis of components like suspensions, attitude controllers, shock absorbers, springs, mounts and steering assemblies as well as brakes and antiskid devices. The complexity of the dynamical equations called for the development of computer-aided formalisms a quarter of a century ago. The theoretical background is today available from a number of textbooks authored by Wittenburg [1], Schiehlen [2], Roberson and Schwertassek [3], Nikravesh [4], Haug [5] and Shabana [6]. The state-of-the-art is also presented at a series of IUTAM/IAVSD symposia, documented in the corresponding proceedings, see, e.g., Magnus [7], Slibar and Springer [8], Haug [9], Kortüm and Schiehlen [10], Bianchi and Schiehlen [11], Kortüm and Sharp [12].

In addition, a number of commercially distributed computer codes were developed, a summary of which is given in the Multibody Systems Handbook [13]. The computer codes available show different capabilities: some of them generate only the equations of motion in numerical or symbolical form, respectively, some of them provide numerical integration and simulation codes, too. Moreover, there are also extensive software systems on the market which offer additionally graphical data input, animation of body motions, and automated signal data analysis. There is no doubt that the professional user, particularly in the automotive industry, prefers the most complete software system for dynamical multibody system analysis.

2.1 Elements of Multibody Systems

The method of multibody systems is based on a finite set of elements such as

- rigid bodies and/or particles,
- bearings, joints, and supports,
- springs and dampers,
- active force and/or position actuators.

For more details see Reference [2] or Roberson and Schwertassek [3]. Each vehicle can be modeled for dynamical analysis by these elements as a multibody system, Figure 3.

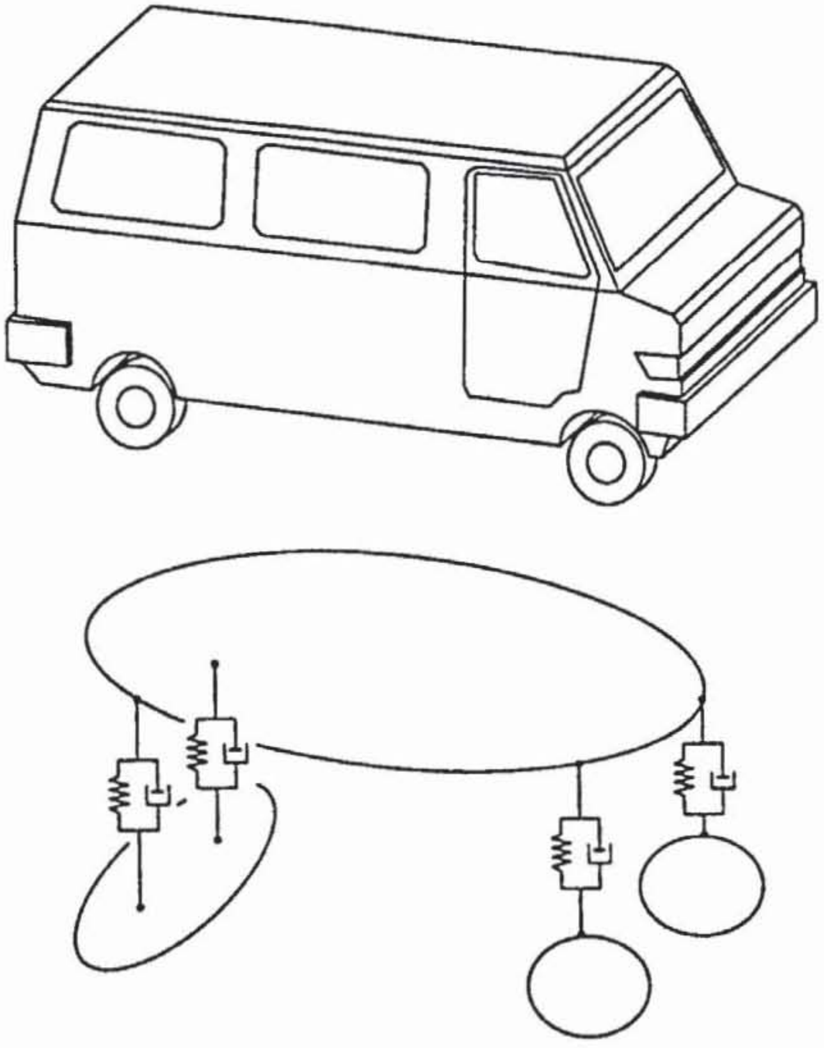


Figure 3. Multibody Model of Vehicle

The elements have to be characterized by body-fixed frames, Figure 4. Then, the absolute and relative motion can be defined by the frame motion using the kinematical quantities of 3x1–translational vectors **r** and 3x3–rotation tensors **S**. The description of joints, Figure 5, requires two frames, one on each of the connected bodies. The joints constrain the relative motion between two rigid bodies and, as a consequence, reaction forces have to be considered. Figure 6 shows the reaction forces $\mathbf{f}^r$ and the reaction $\mathbf{l}^r$ of a revolute joint in a free body diagram. A library of standard joints is shown in Figure 7. For more details see Daberkow [14].

2.2 Multibody System Datamodel

The German Research Council (DFG) sponsored by a nationwide research project the development of a multibody system datamodel, too. In this project, 14 universities and research centers have been engaged and all of them agreed on the datamodel [15].

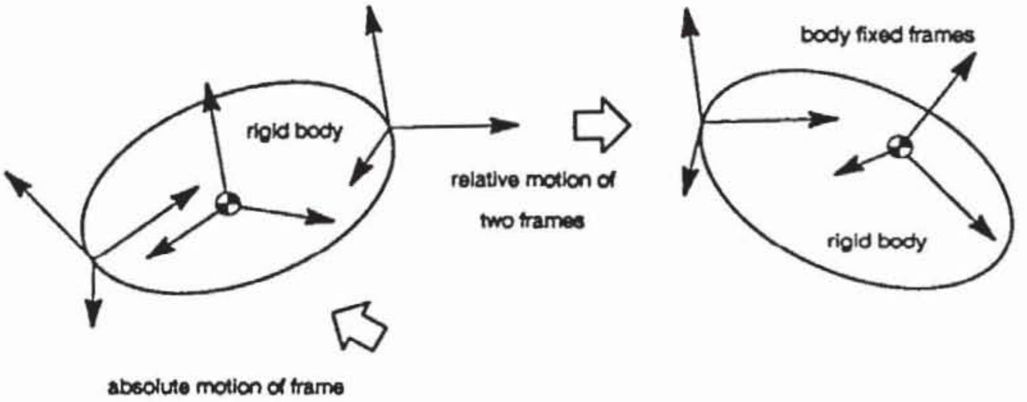


Figure 4. Two Rigid Bodies and Body-Fixed Frames

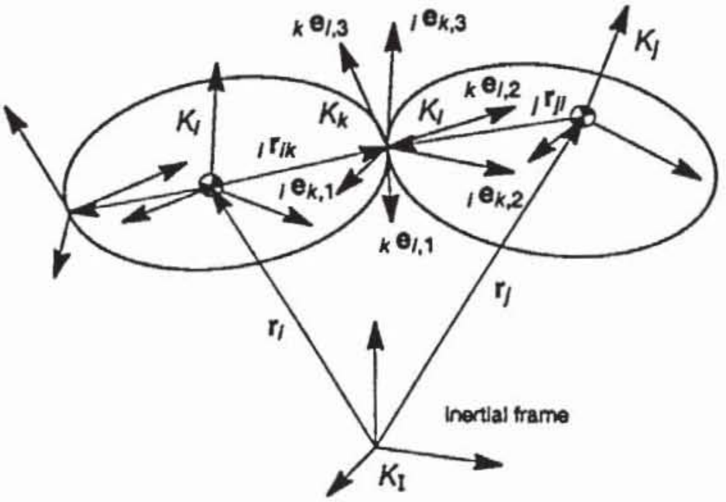


Figure 5. Joint between Two Rigid Bodies

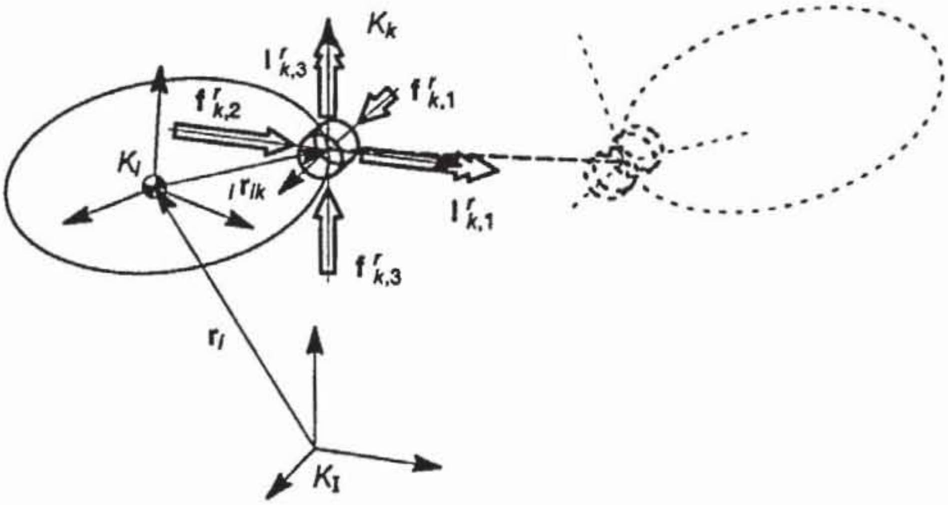


Figure 6. Reaction Forces and Torques of a Revolute Joint

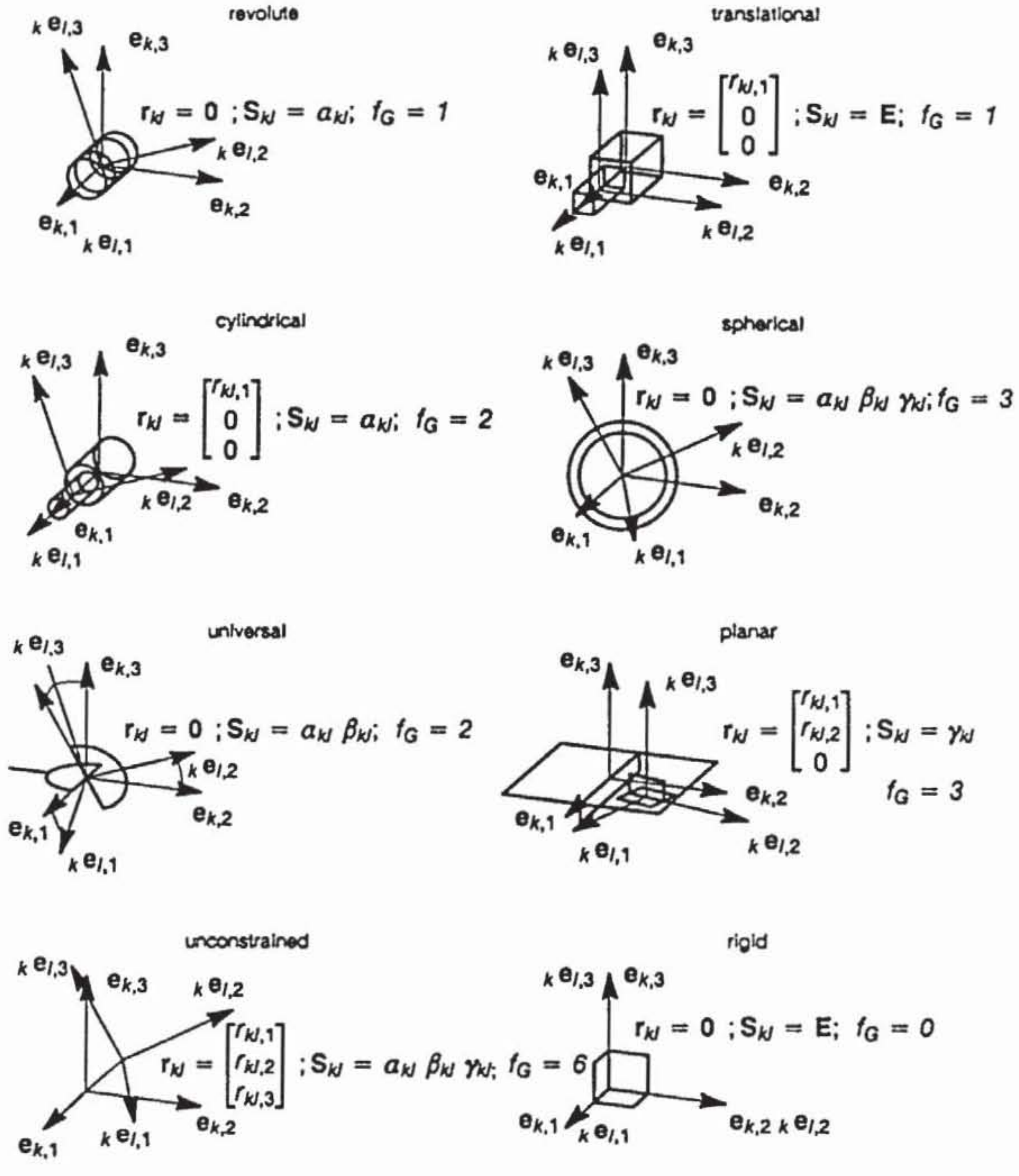

revolute
$_k e_{l,3}$ $e_{k,3}$ $_k e_{l,2}$ $e_{k,2}$ $e_{k,1}$ $_k e_{l,1}$
$r_{kl} = 0 \; ; S_{kl} = \alpha_{kl}; \; f_G = 1$
translational
$_k e_{l,3}$ $e_{k,3}$ $e_{k,1}$ $e_{k,2}$ $_k e_{l,2}$ $_k e_{l,1}$
$r_{kl} = \begin{bmatrix} r_{kl,1} \\ 0 \\ 0 \end{bmatrix}; S_{kl} = E; \; f_G = 1$
cylindrical
$_k e_{l,3}$ $e_{k,3}$ $_k e_{l,2}$ $e_{k,2}$ $e_{k,1}$ $_k e_{l,1}$
$r_{kl} = \begin{bmatrix} r_{kl,1} \\ 0 \\ 0 \end{bmatrix}; S_{kl} = \alpha_{kl}; \; f_G = 2$
spherical
$_k e_{l,3}$ $e_{k,3}$ $_k e_{l,2}$ $e_{k,2}$ $e_{k,1}$ $_k e_{l,1}$
$r_{kl} = 0 \; ; S_{kl} = \alpha_{kl} \; \beta_{kl} \; \gamma_{kl}; f_G = 3$
universal
$_k e_{l,3}$ $e_{k,3}$ $_k e_{l,2}$ $e_{k,2}$ $e_{k,1}$ $_k e_{l,1}$
$r_{kl} = 0 \; ; S_{kl} = \alpha_{kl} \; \beta_{kl}; \; f_G = 2$
planar
$e_{k,3}$ $_k e_{l,3}$ $e_{k,2}$ $e_{k,1}$ $_k e_{l,1}$ $_k e_{l,2}$
$r_{kl} = \begin{bmatrix} r_{kl,1} \\ r_{kl,2} \\ 0 \end{bmatrix}; S_{kl} = \gamma_{kl}$
$f_G = 3$
unconstrained
$_k e_{l,3}$ $e_{k,3}$ $_k e_{l,2}$ $e_{k,1}$ $_k e_{l,1}$ $e_{k,2}$
$r_{kl} = \begin{bmatrix} r_{kl,1} \\ r_{kl,2} \\ r_{kl,3} \end{bmatrix}; S_{kl} = \alpha_{kl} \; \beta_{kl} \; \gamma_{kl}; \; f_G = 6$
rigid
$_k e_{l,3}$ $e_{k,3}$ $e_{k,1}$ $_k e_{l,1}$ $e_{k,2}$ $_k e_{l,2}$
$r_{kl} = 0 \; ; S_{kl} = E; \; f_G = 0$

Figure 7. Library of Standard Joints

The datamodel has been defined as a standardized basis for all kinds of computer codes by Otter, Hocke, Daberkow and Leister [16]. The following assumptions were agreed upon:

1. A multibody system consists of rigid bodies and ideal joints. A body may degenerate to a particle or to a body without inertia. The ideal joints include the rigid joint, the joint with completely given motion (rheonomic constraint) and the vanishing joint (free motion).
2. The topology of the multibody system is arbitrary. Chains, trees and closed loops are admitted.

3. Joints and actuators are summarized in open libraries.
4. Subsystems may be added to existing components of the multibody system.

A datamodel for elastic bodies is under development and will be completely compatible with the rigid body datamodel.

A multibody system as defined is characterized by the class *mbs* and consists of an arbitrary number of the objects of the classes *part* and *interact*, see Figure 8.

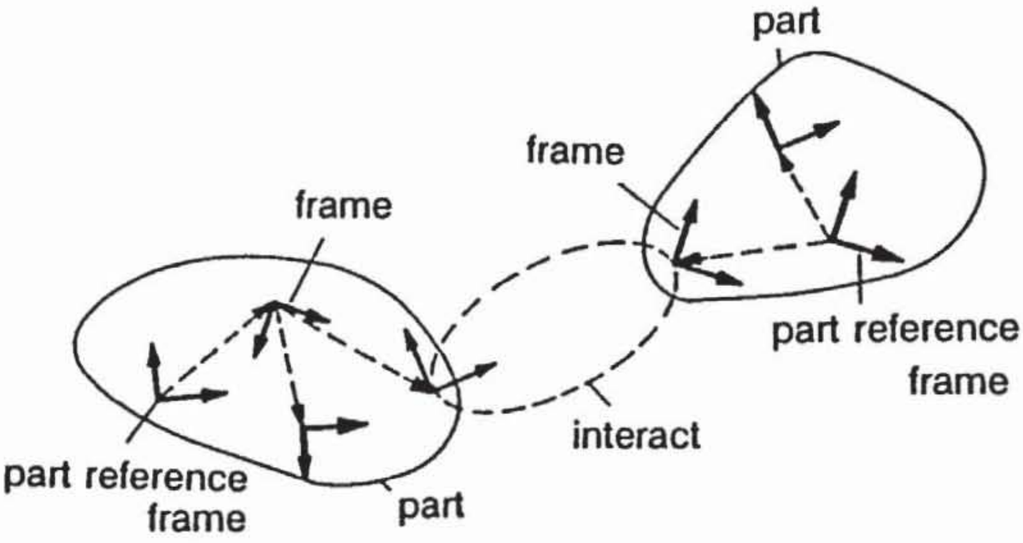


Figure 8. Multibody System to be Represented by the Datamodel

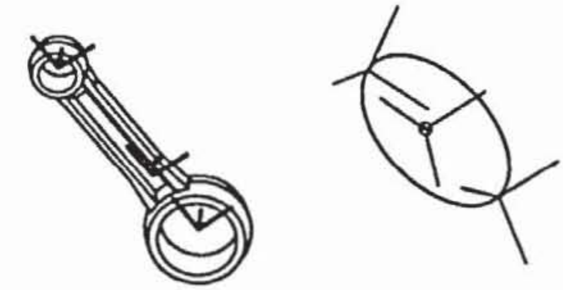
The class *part* describes rigid bodies. Each *part is* characterized by at least one body-fixed *frame*, it may have a mass, a center of mass and a tensor of inertia summarized in the class *body*, Figure 9.

The class *interact* describes the interaction between a frame on *part a* and a frame on *part b*. The interaction may be realized by a joint, by a force actuator or a sensor resulting in the classes *joint, force* or *sensor*, respectively. Thus, the class *interact is* characterized by two types of information: the frames to be connected and the connecting element itself, see Figure 10.

As an example for an element of the class *joint* and the class *force* Figure 11 shows a damper.

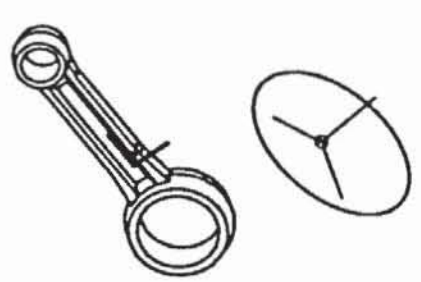
The presented classes are the basis of the class *mbs* which means the assembled vehicle. The model assembly using the datamodel is now easily executed. Figure 12 shows the whole procedure. According to the definitions, the datamodel represents holonomic, rheonomic multibody systems.

Technical system | multibody model | class description



class: part		
name	class	description
body	body	mass properties of part
frame	FRAME	frames on the part

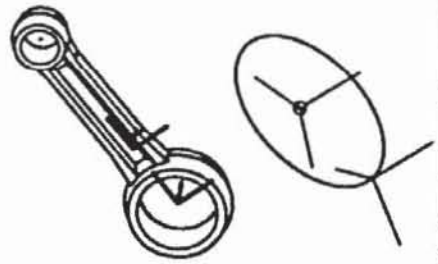
Technical system | multibody model | class description | object of class



class: body		
name	type	description
mass	dparam	mass of body
mframe	name	c.o.g frame
inertia	dparam(6)	inertia tensor
iframe	name	tensor frame

object: body1	
name	component
mass	M1
	0.418
mframe	' '
inertia	I11,I22,I33,I12,I13,I23
	113.0,1296.49,1380.67
iframe	' '

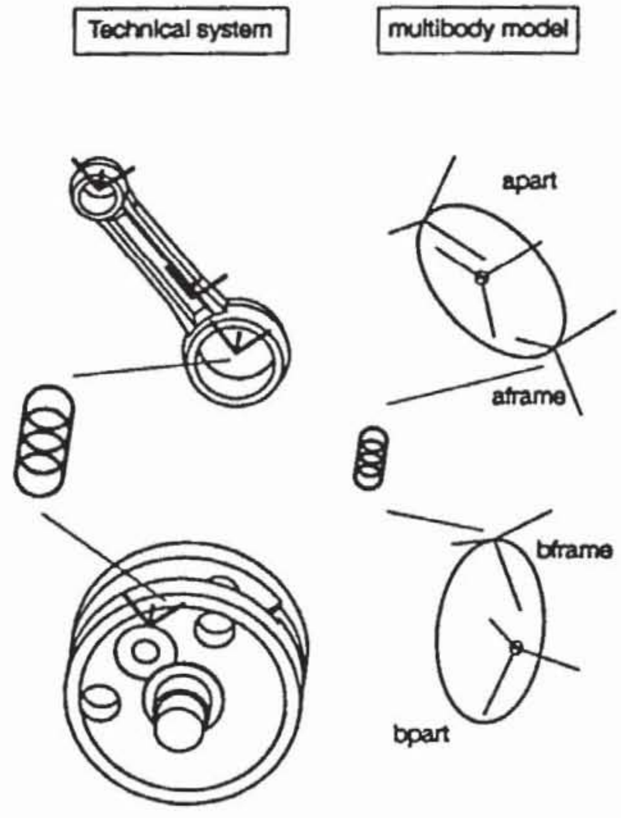
Technical system | multibody model | class description | object of class



class: frame		
name	type	description
rframe	name	reference fr.
origin	dparam(3)	frame origin
axieseq	int(3)	rot. sequence
rangles	dparam(3)	rotation angles

object: frame1	
name	component
rframe	' '
origin	L1,L2,L3
	-44.21,0.0,0.0
axieseq	1,2,3
rangles	AL,BE,GA
	0.0,0.0,0.0

Figure 9. Definition of Class *Part*



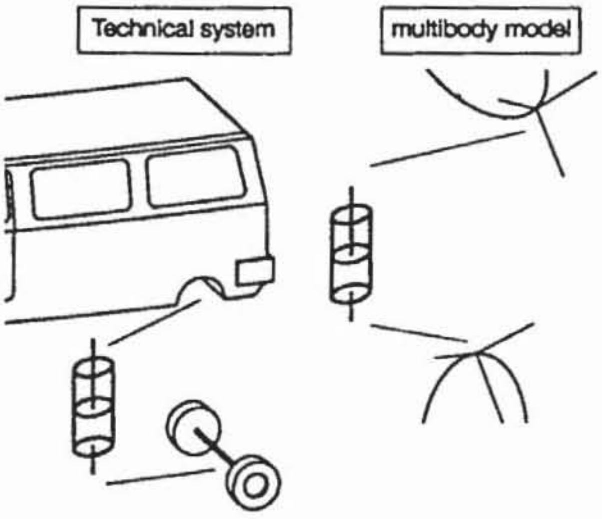
class description

class: Interact		
name	class	description
connect	connect	frames to be connected
member	member	connecting element

class: connect		
name	class	description
apart	name	name of apart
aframe	name	name of aframe on apart
bpart	name	name of bpart
bframe	name	name of bframe on bpart

class: member		
name	class	description
joint	joint	joint element on interact
force	FORCE	force elements on i'act
sensor	SENSOR	sensor elements on i'act

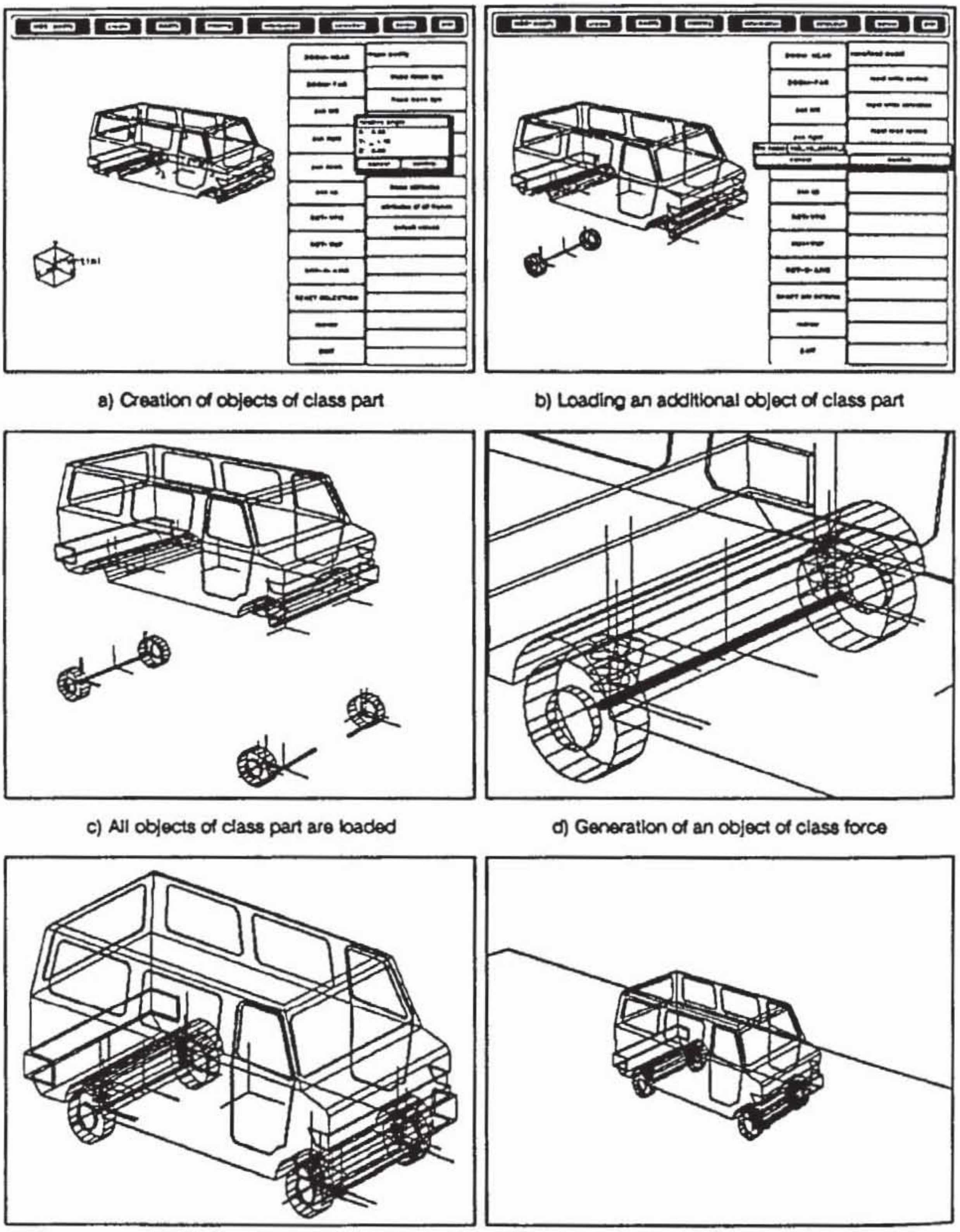
Figure 10. Definition of Class *Interact*



class description

class: dampert		
name	type	description
subclass of (force)		
d	param	damping coefficient

Figure 11. Definition of Class *Dampert*



a) Creation of objects of class part

b) Loading an additional object of class part

c) All objects of class part are loaded

d) Generation of an object of class force

e) assembled object of class mbs

f) Exchange of graphic entity for inertial part object

Figure 12. Model Assembly of a Van

2.3 The Database RSYST

The scientific-engineering database system RSYST is chosen as a software tool. It supports the

- development of user programs,
- development of program packages,
- execution of programs,
- handling of large data sets,
- analysis of data.

One of the main applications of RSYST is the compilation of data and user programs. RSYST is written in FORTRAN 77 and, therefore, it has an excellent portability to all kinds of computers. The fundamental elements of RSYST are the following:

- execution control,
- information system,
- dialogue system,
- output handling,
- dynamic storage handling,
- method and model base,
- database.

The RSYST system has been developed by Rühle and his staff at the Computing Center of the University of Stuttgart. A detailed description is given by Lang [17], Loebich [18] and Rühle [19].

Most important for the multibody system datamodel are the RSYST database and the handling of data objects. All the data in RSYST are considered as objects of a database. Such data objects are, for example, vectors, matrices, sets of parameters, texts, or formally defined objects. The data objects are stored in the RSYST database subject to a very efficient handling, they are identified by special names.

Each data object in the RSYST database is characterized by a data description, identifying the data type. The data description permits a correlation between data objects and possible operations.

RSYST offers the following operations on data objects which are completely internally executed:

- object generating,
- object changing,
- object deleting,
- object listing,
- objects relating to each other,

- objects storing and reading,
- handling of components of objects.

The objects have to be interpreted for the identification of their information. A set of objects with same rules of interpretation are called a class specified by a name. For example, the four components *"rframe, origin, axleseq, rangles"* of the class *"frame,"* are shown in Figure 9.

The database RSYST offers a suitable software engineering concept for multibody system dynamics as shown in Figure 13.

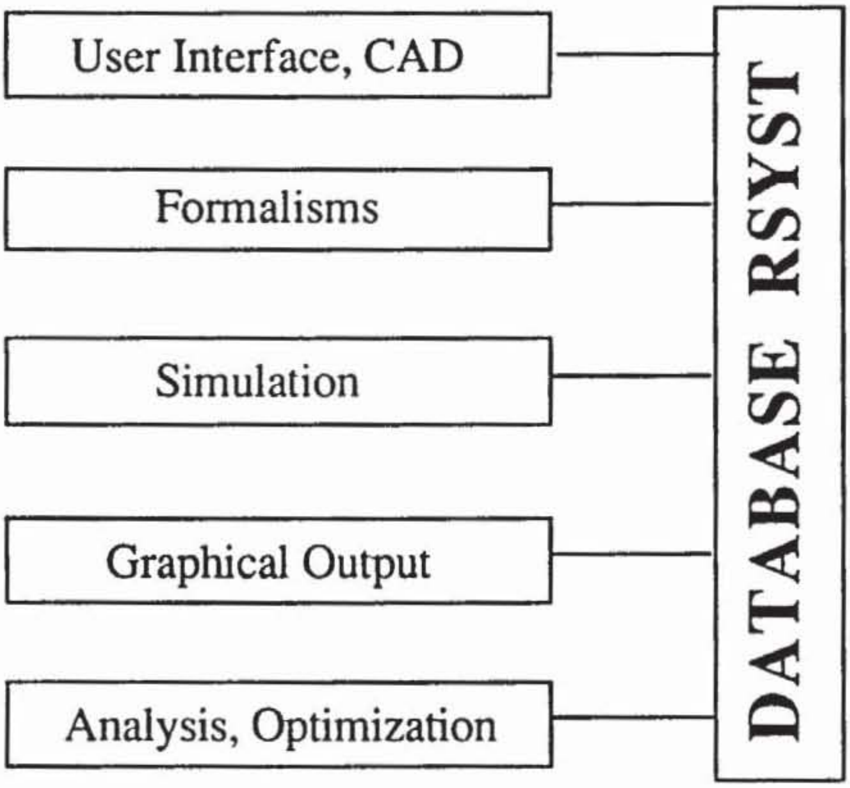
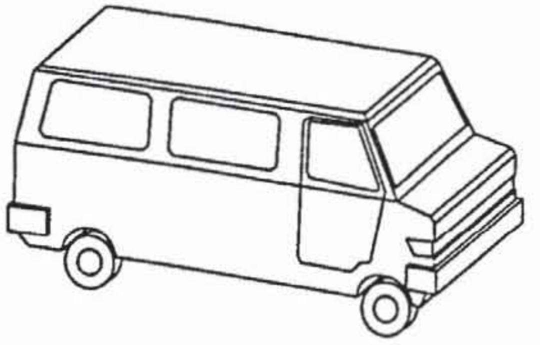


Figure 13. Software Engineering Concept of Multibody System Dynamics

This concept provides the opportunity to use a modular structure of the software, i.e., different multibody formalisms may be combined with different simulation programs via standardized interfaces.

The database structure of the assembled van presented by Figure 12 is shown in Figure 14. It is obvious that the two identical front wheel assemblies result in identical data input. In particular, different designs can be easily implemented and tested.

The object-oriented multibody system modeling can be included in any CAD-3D-software by minor adjustments. The object-oriented multibody modeling kernel, shown in Figure 15, handles the data input and dialog the data output, the storage of *mbs* data and the 3D-graphics. As a result, the CAD-3D and CAD-2D model files are supplemented by a multibody model file.



```
PART
   _inertial
      _FRAME
         _iframe      frame iframe of part inertial
   _vanbody
      _BODY           van-body
      _FRAME
         _jlfwheel    frame of joint left front wheel of part
                      van-body
         _jrfwheel    frame of joint right front wheel of part
                      van-body
         _jraxle      frame of joint rearaxle of part van-body
         _sdlfw       frame of spring-damperforce left front wheel
         _sdrfw       frame of spring-damperforce right front wheel
         _slra        frame of spring rearaxle left
         _dlra        frame of damper rearaxle left
         _srra        frame of spring rearaxle right
         _drra        frame of damper rearaxle right
   _lfrontw
      _BODY           left front wheel -----------------------+
      _FRAME                                                  I
         _jvanbody    frame of joint vanbody of part left     I
                      front wheel ----------------------------I-+
         _sdvb        frame of spring-damperforce vanbody  ---I-I-+
         _swheel      frame of springforce left front wheel -I-I-I-+
   _rfrontw                                                   I I I I
      _BODY           right front wheel ----------------------+ I I I
      _FRAME                                                    I I I
         _jvanbody    frame of joint vanbody of part right      I I I
                      front wheel ------------------------------+ I I
         _sdvb        frame of spring-damperforce vanbody  -------+ I
         _swheel      frame of springforce right front wheel  ------+
   _rearaxle
      _BODY           rearaxle
      _FRAME
         _bframe      frame bodyframe of part rearaxle
         _jvanbody    frame of joint vanbody of part right front
                      wheel
         _slvb        frame of left springforce vanbody
         _dlvb        frame of left damperforce vanbody
         _srvb        frame of right springforce vanbody
         _drvb        frame of right damperforce vanbody
         _slwheel     frame of springforce right rear wheel
         _srwheel     frame of springforce left rear wheel
```

Figure 14. Database Structure of a Van

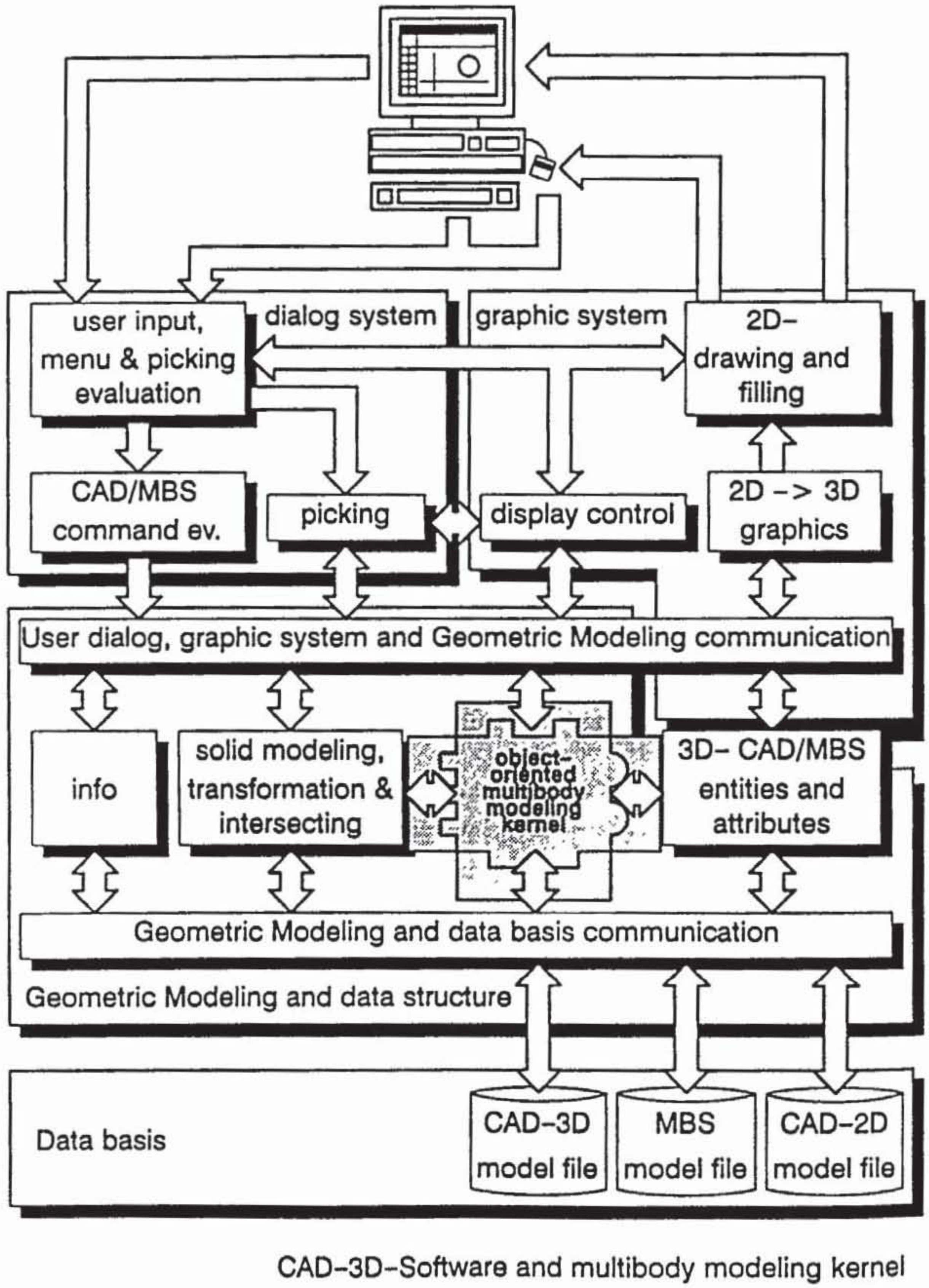


Figure 15. CAD Software and Multibody Modeling Kernel

3 Generation of Equations of Motion

The multibody system model has to be described mathematically by equations of motions for the dynamical analysis. In this chapter, the general theory for holonomic and nonholonomic systems will be presented using a minimal number of generalized coordinates for a unique representation of the motion.

3.1 Kinematics of Multibody Systems

According to the free body diagram of a vehicle system, firstly, all constraints are omitted and the system of p bodies holds 6p degrees of freedom. The position of the system is given relative to the inertial frame by the 3x1–translation vector

$$\mathbf{r}_i = \begin{bmatrix} r_{i1} & r_{i2} & r_{i3} \end{bmatrix}^T, i = 1(1)p, \tag{1}$$

of the center of mass C_i and the 3x3–rotation tensor

$$\mathbf{S}_i = \mathbf{S}_i(\alpha_i \; \beta_i \; \gamma_i)^T, \tag{2}$$

written down for each body. The rotation tensor $\mathbf{S}_i$ depends on three angles α_i, β_i, γ_i and corresponds with the direction cosine matrix relating the inertial frame I and the body-fixed frame i to each other. The 3p translational coordinates and the 3p rotational coordinates (angles) can be summarized in a 6pxl-position vector

$$\mathbf{x} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{21} & \cdots & \alpha_p & \beta_p & \gamma_p \end{bmatrix}^T. \tag{3}$$

Equations (1) and (2) now read

$$\mathbf{r}_i = \mathbf{r}_i(\mathbf{x}), \; \mathbf{S}_i = \mathbf{S}_i(\mathbf{x}) \; . \tag{4}$$

Secondly, the q holonomic, rheonomic constraints are added to the vehicle system given explicitly by

$$\mathbf{x} = \mathbf{x}(\mathbf{y}, t), \tag{5}$$

where the fx1–position vector

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_f \end{bmatrix}^T \tag{6}$$

is used summarizing the f generalized coordinates of the system. The number of generalized coordinates corresponds to the number of degrees of freedom, f = 6p–q, with respect to the systems position. Then, translation and rotation of each body follow from (4) and (5) as

$$\mathbf{r}_i = \mathbf{r}_i(\mathbf{y},t),\ \mathbf{S}_i = \mathbf{S}_i(\mathbf{y},t) \tag{7}$$

and the velocities are found by differentiation with respect to the inertial frame:

$$\mathbf{v}_i = \dot{\mathbf{r}}_i = \frac{\partial \mathbf{r}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \mathbf{r}_i}{\partial t} = \mathbf{J}_{Ti}(\mathbf{y},t)\ \dot{\mathbf{y}} + \overline{\mathbf{v}}_i(\mathbf{y},t), \tag{8}$$

$$\boldsymbol{\omega}_i = \dot{\mathbf{s}}_i = \frac{\partial \mathbf{s}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \mathbf{s}_i}{\partial t} = \mathbf{J}_{Ri}(\mathbf{y},t)\ \dot{\mathbf{y}} + \overline{\boldsymbol{\omega}}_i(\mathbf{y},t). \tag{9}$$

The 3xf–Jacobian matrices $\mathbf{J}_{Ti}$ and $\mathbf{J}_{Ri}$ defined by (8) and (9) characterize the virtual translational and rotational displacement of the system, respectively. They are also needed later for the application of d'Alembert's principle. The infinitesimal 3x1–rotation vector $\mathbf{s}_i$ used in (9) follows analytically from the corresponding infinitesimal skew–symmetrical 3x3–rotation tensor. However, the matrix $\mathbf{J}_{Ri}$ can also be found by a geometrical analysis of the angular velocity vector $\boldsymbol{\omega}_i$ with respect to the angles α_i, β_i, γ_i, see e.g., Reference [2].

The accelerations are obtained by a second differentiation with respect to the inertial frame:

$$\mathbf{a}_i = \mathbf{J}_{Ti}(\mathbf{y},t)\ \ddot{\mathbf{y}} + \frac{\partial \mathbf{v}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \mathbf{v}_i}{\partial t}, \tag{10}$$

$$\boldsymbol{\alpha}_i = \mathbf{J}_{Ri}(\mathbf{y},t)\ \ddot{\mathbf{y}} + \frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \boldsymbol{\omega}_i}{\partial t}. \tag{11}$$

For scleronomic constraints, the partial time-derivatives in (8), (9) and (10), (11) vanish.

Thirdly, the r nonholonomic, rheonomic constraints, especially due to rigid wheels, are introduced explicitly by

$$\dot{\mathbf{y}} = \dot{\mathbf{y}}(\mathbf{y},\mathbf{z},t) \tag{12}$$

with the gx1–velocity vector

$$\mathbf{z}(t) = \left[z_1\ z_2\ z_3\ \cdots\ z_g\right]^T \tag{13}$$

summarizing the g generalized velocities of the system. The number of generalized velocities characterizes the number of degrees of freedom, g = f – r, with respect to the system's velocity. From (8), (9), and (12), the translational and rotational velocity of each body follow immediately as

$$\mathbf{v}_i = \mathbf{v}_i(\mathbf{y},\mathbf{z},t), \ \boldsymbol{\omega}_i = \boldsymbol{\omega}_i(\mathbf{y},\mathbf{z},t). \tag{14}$$

The accelerations are found again by differentiation with respect to inertial frame I:

$$\mathbf{a}_i = \frac{\partial \mathbf{v}_i}{\partial \mathbf{z}}\dot{\mathbf{z}} + \frac{\partial \mathbf{v}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \mathbf{v}_i}{\partial t} = \mathbf{L}_{Ti}(\mathbf{y},\mathbf{z},t)\ \dot{\mathbf{z}} + \bar{\bar{\mathbf{v}}}_i(\mathbf{y},\mathbf{z},t), \tag{15}$$

$$\boldsymbol{\alpha}_i = \frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{z}}\dot{\mathbf{z}} + \frac{\partial \boldsymbol{\omega}_i}{\partial \mathbf{y}}\dot{\mathbf{y}} + \frac{\partial \boldsymbol{\omega}_i}{\partial t} = \mathbf{L}_{Ri}(\mathbf{y},\mathbf{z},t)\ \dot{\mathbf{z}} + \dot{\bar{\bar{\boldsymbol{\omega}}}}_i(\mathbf{y},\mathbf{z},t). \tag{16}$$

Here, the 3xg–matrices $\mathbf{L}_{Ti}$ and $\mathbf{L}_{Ri}$ are introduced for the description of the virtual translational and rotational velocity of the system needed also for the application of Jourdain's principle. Further, it has to be mentioned that the partial time–derivatives vanish in (15), (16) for scleronomic systems.

In many applications, a reference frame is given in a natural way. For example, a railway vehicle running on a curved super-elevated track is naturally described in a moving track-related frame. Therefore, the absolute motion may be also presented in a reference frame using the reference motion itself and the bodies' relative motion [20].

3.2 Newton-Euler Equations

For the application of Newton's and Euler's equation to multibody systems, the free body diagram has to be used again. Now the rigid bearings and supports are replaced by adequate constraint forces and torques as discussed later in this section.

Newton's and Euler's equation read for each body in the inertial frame

$$m_i\dot{\mathbf{v}}_i = \mathbf{f}_i^{e} + \mathbf{f}_i^{r}, \quad i = 1(1)p, \tag{17}$$

$$\mathbf{I}_i\dot{\boldsymbol{\omega}}_i + \tilde{\boldsymbol{\omega}}_i\mathbf{I}_i\boldsymbol{\omega}_i = \mathbf{l}_i^{e} + \mathbf{l}_i^{r}, \quad i = 1(1)p. \tag{18}$$

The inertia is represented by the mass m_i and the 3x3–inertia tensor $\mathbf{I}_i$ with respect to the center of mass C_i of each body. The external forces and torques in (17) and (18) are composed by the 3x1–applied force vector $\mathbf{f}_i^{e}$ and torque vector $\mathbf{l}_i^{e}$ due to springs, dampers, actuators, weight, etc., and by the 3x1–constraint force vector $\mathbf{f}_i^{r}$ and torque vector $\mathbf{l}_i^{r}$. All torques are related to the center mass C_i . The applied forces and torques, respectively, depend on the motion by different laws, and they may be coupled to the constraint forces and torques in the case of friction.

The constraint forces and torques originate from the reactions in joints, bearings, supports, or wheels. They can be reduced by distribution matrices to the generalized constraint forces. The number of the generalized constraint forces is equal to the total number of constraints (q+r) in the system. Introducing the (q+r)x1–vector of generalized constraint forces

$$\mathbf{g} = \left[g_1\ g_2\ g_3\ \cdots\ g_{q+r}\right]^T \tag{19}$$

and the 3x(q+r)–distribution matrices

$$\mathbf{F}_i = \mathbf{F}_i(\mathbf{y},\mathbf{z},t), \quad \mathbf{L}_i = \mathbf{L}_i(\mathbf{y},\mathbf{z},t) \tag{20}$$

it turns out

$$\mathbf{f}_i{}' = \mathbf{F}_i\mathbf{g}, \quad \mathbf{1}_i{}' = \mathbf{L}_i\mathbf{g}, \quad i = 1(1)p, \tag{21}$$

for each body. The constraint forces or the distribution matrices, respectively, can be found mathematically, or they are derived by geometrical analysis.

The ideal applied forces and torques depend only on the kinematical variables of the system, they are independent of the constraint forces. Ideal applied forces are due to the elements of multibody systems and further actions on the system, e.g., gravity. The forces may be characterized by proportional, differential, and/or integral behavior.

The proportional forces are characterized by the system's position and time-functions

$$\mathbf{f}_i^e = \mathbf{f}_i^e(\mathbf{x},t). \tag{22}$$

For example, conservative spring and weight forces, as well as purely time-varying forces, are proportional forces.

The proportional-differential forces depend on the position and the velocity:

$$\mathbf{f}_i^e = \mathbf{f}_i^e(\mathbf{x},\dot{\mathbf{x}},t). \tag{23}$$

A parallel spring-dashpot configuration is a typical example for this class of forces. The proportional-integral forces are a function of the position and integrals of the position:

$$\mathbf{f}_i^e = \mathbf{f}_i^e(\mathbf{x},\mathbf{w},t), \quad \dot{\mathbf{w}} = \dot{\mathbf{w}}(\mathbf{x},\mathbf{w},t)\ , \tag{24}$$

where the px1–vector w describes the position integrals. For example, serial spring-damper configurations and the eigendynamics of actuators result in proportional-integral forces. In vehicle systems proportional-integral forces appear, e.g., with modem engine mounts for simultaneous noise and vibration reduction. The same laws hold also for ideal applied torques.

In the case of non ideal constraints with sliding friction or contact forces, respectively, the applied forces are coupled with the constraint forces [20].

The Newton-Euler equations of the complete system are summarized in matrix notation by the following vectors and matrices. The inertia properties are written in the 6px6p–diagonal matrix

$$\overline{\mathbf{M}} = \mathrm{diag}\left\{ m_1\mathbf{E} \quad m_2\mathbf{E} \cdots \mathbf{I}_1 \cdots \mathbf{I}_p \right\}, \tag{25}$$

where the 3x3–identity matrix $\mathbf{E}$ is used. The 6px1–force vectors $\overline{\mathbf{q}}^c$, $\overline{\mathbf{q}}^e$, $\overline{\mathbf{q}}^r$ representing the coriolis forces, the ideal applied forces and the constraint forces, respectively, are given by the following scheme,

$$\overline{\mathbf{q}} = \left[\mathbf{f}^T{}_1 \, \mathbf{f}^T{}_2 \; \cdots \; \mathbf{l}^T{}_1 \; \cdots \; \mathbf{l}^T{}_P \right]^T. \tag{26}$$

Further the 6pxf–matrix $\overline{\mathbf{J}}$ and 6pxg–matrix $\overline{\mathbf{L}}$ as well as the 6px(q+r)–distribution matrix $\overline{\mathbf{Q}}$ are introduced as global matrices, e.g.,

$$\overline{\mathbf{J}} = \left[\mathbf{J}_{T1}{}^T \, \mathbf{J}_{T2}{}^T \; \cdots \; \overline{\mathbf{J}}_{R1}{}^T \; \cdots \; \overline{\mathbf{J}}_{RP}{}^T \right]^T. \tag{27}$$

Now, the Newton-Euler equations can be represented as follows for holonomic systems in the inertial frame

$$\overline{\mathbf{M}}\,\overline{\mathbf{J}}\,\ddot{\mathbf{y}} + \overline{\mathbf{q}}^c(\mathbf{y},\dot{\mathbf{y}},t) = \overline{\mathbf{q}}^e(\mathbf{y},\dot{\mathbf{y}},t) + \overline{\mathbf{Q}}\,\mathbf{g} \tag{28}$$

and for nonholonomic systems

$$\overline{\mathbf{M}}\,\overline{\mathbf{L}}\,\dot{\mathbf{z}} + \overline{\mathbf{q}}^c(\mathbf{y},\mathbf{z},t) = \overline{\mathbf{q}}^e(\mathbf{y},\mathbf{z},t) + \overline{\mathbf{Q}}\,\mathbf{g}. \tag{29}$$

If the holonomic constraints are omitted, e.g., $\mathbf{z} = \dot{\mathbf{y}}$, Equation (29) reduces to (28), showing a close relation between both representations.

3.3 Equations of Motion

The Newton-Euler equations are combined algebraical and differential equations and the question arises if they can be separated for solution into purely algebraical and differential equations. There is a positive answer given by the dynamical principles. In a first step, the system's motion can be found by integration of the separated differential equations and in a second step the constraint forces are calculated algebraically. For ideal applied forces, both steps can be executed successively while contact forces require simultaneous execution.

Holonomic systems with proportional or proportional-differential forces result in *ordinary* multibody systems. The equations of motion follow from the Newton-Euler equations, applying d'Alembert's principle.

The equations of motion of holonomic systems are found according to d'Alembert's principle by premultiplication of (28) with $\overline{\mathbf{J}}^{\mathrm{T}}$ as

$$\mathbf{M}(\mathbf{y},t)\,\ddot{\mathbf{y}}+\mathbf{k}(\mathbf{y},\dot{\mathbf{y}},t)=\mathbf{q}(\mathbf{y},\dot{\mathbf{y}},t). \tag{30}$$

Here, the number of equations is reduced from 6p to f, the fxf–inertia matrix $\mathbf{M}(\mathbf{y},t)$ is completely symmetrized $\mathbf{M}(\mathbf{y},t) = \overline{\mathbf{J}}^{\mathrm{T}}\;\overline{\overline{\mathbf{M}}}\,\overline{\mathbf{J}} > 0$, and the constraint forces and torques are eliminated. The remaining fxl–vector $\mathbf{k}$ describes the generalized coriolis forces and the fx1–vector $\mathbf{q}$ includes the generalized applied forces.

The equations of motion following from a moving reference frame agree completely with (30). Thus, the choice of the reference frame doesn't affect the equations of motion at all. However, kinematics and Newton-Euler equations as well as the application of d'Alembert's principle may be strongly simplified by the choice of proper reference frames.

Nonholonomic systems with proportional-integral forces produce *general* multibody systems. The equations of motion are obtained from the Newton-Euler Equations (29) where the proportional-integral forces (24) and Jourdain's principle have to be regarded. However, the equations of motion are not sufficient, they have to be completed by the nonholonomic constraint equation (12). Thus, the complete equations read as

$$\begin{aligned}&\mathbf{M}(\mathbf{y},\mathbf{z},t)\dot{\mathbf{z}}+\mathbf{k}(\mathbf{y},\mathbf{z},t)=\mathbf{q}(\mathbf{y},\mathbf{z},\mathbf{w},t),\\&\dot{\mathbf{y}}=\dot{\mathbf{y}}(\mathbf{y},\mathbf{z},t),\quad \dot{\mathbf{w}}=\dot{\mathbf{w}}(\mathbf{y},\mathbf{z},t).\end{aligned} \tag{31}$$

Now, the number of equations is reduced from 6p to g and the gxg–symmetric inertia matrix $\mathbf{M}(\mathbf{y},\mathbf{z},t)=\overline{\mathbf{L}}^{\mathrm{T}}\;\overline{\overline{\mathbf{M}}}\,\overline{\mathbf{L}}>0$ appears. Further, $\mathbf{k}$ and $\mathbf{q}$ are gx1–vectors of generalized coriolis and applied forces. The Equations (31) are in the literature also denoted as Kane's equations.

In addition to the mechanical representation (31) of a multibody system, there also exists the possibility to use the more general representation of dynamical systems, e.g.,

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},\mathbf{u},t,\mathbf{p}),\;,\quad \mathbf{v}=\mathbf{g}(\mathbf{x},\mathbf{u},t,\mathbf{p}), \tag{32}$$

where $\mathbf{x}$ means in (33) the state vector, $\mathbf{v}$ the output vector, $\mathbf{u}$ the input vector of controls, t the time and $\mathbf{p}$ the vector of mechanical and control parameters or design variables, respectively.

The constraint forces are completely omitted by the dynamical principles. However, they are also of engineering interest for the load in joints, bearings and supports, and they are absolutely necessary for the computation of contact and friction forces. From the 6p coordinates of the constraint force vector $\overline{\mathbf{q}}^{r}$ there are only (q+r) coordinates linear independent according to (21). Therefore, only the (q+r)x1–vector $\mathbf{g}$ of the generalized constraint forces is needed. The results are given for holonomic systems only, r=0, but they can be transferred to nonholonomic systems without any problem.

The premultiplication of (28) by $\overline{\mathbf{Q}}^{\mathrm{T}}\ \overline{\mathbf{M}}^{-1}$ results according to d'Alembert's principle immediately in the equations of reaction

$$\mathbf{N}(\mathbf{y},t)\ \mathbf{g} + \hat{\mathbf{q}}(\mathbf{y},\dot{\mathbf{y}},t) = \hat{\mathbf{k}}(\mathbf{y},\dot{\mathbf{y}},t) \tag{33}$$

where $\mathbf{N}(\mathbf{y},t) = \overline{\mathbf{Q}}^{\mathrm{T}}\ \overline{\mathbf{M}}^{-1}\ \overline{\mathbf{Q}} > 0$ is the symmetrical qxq–reaction matrix and $\hat{\mathbf{q}}$ and $\hat{\mathbf{k}}$ are qxl–vectors.

3.4 Formalism NEWEUL

The equations of motion presented are automatically generated by the formalism NEWEUL described in the Multibody Systems Handbook [13], too.

NEWEUL is a software package for the dynamic analysis of mechanical systems with the multibody system method. It comprises the computation of the symbolic equations of motion by the modul NEWEUL and the simulation of the dynamic behavior by the modul NEWSIM. Multibody systems are mechanical models consisting of

- rigid bodies,
- arbitrary constraining elements (joints, position control elements),
- passive coupling elements (springs, dampers), and
- active coupling elements (force control elements).

The topological structure of the models is arbitrary, thus possible configurations are

- systems with chain structure,
- systems with tree structure, and
- systems with closed kinematical loops.

The scieronomic or rheonomic constraints may be

- holonomic or
- nonholonomic.

The software package NEWEUL has been successfully applied in industrial and academic research institutions since 1979. The major fields of application are

- vehicle dynamics,
- dynamics of machinery,
- robot dynamics,
- biomechanics,
- satellite dynamics,
- dynamics of mechanisms.

The software package NEWEUL offers two approaches for multibody system modeling.

These are

- the successive assembly approach using the kinematics of relative motions, and
- the modular assembly approach based on subsystems.

The input data for NEWEUL have to be entered in input files prepared with prompts and comments.

4 Numerical Simulation of Automotive Systems

For the simulation of multibody systems, the computational efficiency is most important. A drawback of the equations of motion (30) is the fact that the inertia matrix has to be inverted

$$\ddot{\mathbf{y}} = \mathbf{M}^{-1}(\mathbf{q} - \mathbf{k}) \tag{34}$$

a procedure of high computational costs. In robot dynamics, however, there has been developed quite a number of methods to circumvent this difficulty see Hollerbach [21], Walker and Orin [22], Brandl, Johanni and Otter [23]. It turns out that for open chains, the second derivative of (5) can be replaced by the recursive kinematic relation

$$\ddot{\mathbf{x}} = \mathbf{C}\ddot{\mathbf{x}} + \mathbf{J}\ddot{\mathbf{y}} + \boldsymbol{\xi} \tag{35}$$

and the Newton-Euler Equations (28) are to be rewritten as

$$\overline{\mathbf{M}}\ddot{\mathbf{x}} + \overline{\mathbf{k}} = \overline{\mathbf{q}^{e}} + (\mathbf{E} - \mathbf{C}^{T})\mathbf{Q}\mathbf{g} \tag{36}$$

where **C** is a 6px6p–geometry-matrix with submatrices on the lower subdiagonal only and **J** and **Q** are blockdiagonal matrices of local Jacobian and distribution matrices [24]. Further, the 6px6p–inertia matrix $\overline{\mathbf{M}}$ is block-diagonal and time-invariant. Equations (35) and (36) can be solved recursively, avoiding the inversion of the fxf–inertia matrix **M** from (30). In Reference [24], it has been shown that the recursive formalisms can be interpreted as a sophisticated backward recursion using the Gaussian algorithm. However, all the recursive approaches are restricted to chain or tree topology of multibody systems, closed kinematical loops cannot be treated.

Equation (35) and (36) represent 12p differential algebraical equations (DAE) for the 12p unknowns summarized in the vector x, y, g where p is the number of bodies, see Chapter 3. The elimination of the relative accelerations $\ddot{\mathbf{y}}$ in (35) by premultiplication with $\mathbf{Q}^{T}$ according to the orthogonality condition results in

$$\ddot{\Phi}(\mathbf{x},\dot{\mathbf{x}},\ddot{\mathbf{x}}) = \mathbf{Q}^{T}(\mathbf{E} - \mathbf{C}^{T})\,\ddot{\mathbf{x}} - \mathbf{Q}^{T}\boldsymbol{\xi} \tag{37}$$

a set of q equations which have the second integral

$$\Phi(\mathbf{x}) = \mathbf{0} \tag{38}$$

representing the q holonomic, rheonomic constraints (5) of the multibody system implicity. The elimination of the absolute accelerations $\ddot{\mathbf{x}}$ in (35) and (36) results again in the f equations of motion (30).

4.1 Computational Aspects of Simulation

The partially reduced Equations (36) and (38) represent differential algebraical equations (DAE) with the 6p+q unknowns in vectors **x**, **g** and the sparse matrices $\overline{\mathbf{M}}$, **C** and **Q**. On the other hand, the fully reduced equations (30) represent ordinary differential equations (ODE) with 6p–q = f unknowns in vector **y** and the full matrix **M**. There are several integration codes for DAE and ODE available which have been tested by Leister [25]. The results presented in Figure 16 show that for closed and open loop structures, the full reduced equations are more efficient. In the case of open loops, the ODE approach can be speeded up recursively if the number of bodies is p>8 as Valasek [26] found.

Due to the implicit formulation of the constraint equations, the closed loop problem does not exist in the DAE approach. However, in the ODE approach it deserves a special treatment. There are two possibilities to overcome this problem:

- automatic choice of an optimal set of generalized coordinates during integration as implemented by Leister [27] and
- preselection of complementary sets of generalized coordinates by kinematical analysis before integration as proposed in Reference [24].

The first method is used in the latest version of the formalism NEWEUL.

4.2 Dynamical Analysis with NEWEUL-NEWSIM Software

NEWEUL generates the equation of motion of multibody systems in symbolic form. The computation is based on the Newton-Euler approach with application of the principles of d'Alembert and Jourdain. The resulting equations of motion may be

- linear,
- partially linearized, or
- nonlinear

symbolic differential equations. Constant parameters can be included in numerical form. Nonlinear coupling elements in kinematically linear models are also permitted.

For the output format of the equations of motion, several options are possible. FORTRAN compatible output allows the equations to be included in commercial software packages for dynamic analysis and simulation such as, ACSL. Another output format allows the processing of the equations with the formula manipulation program MAPLE.

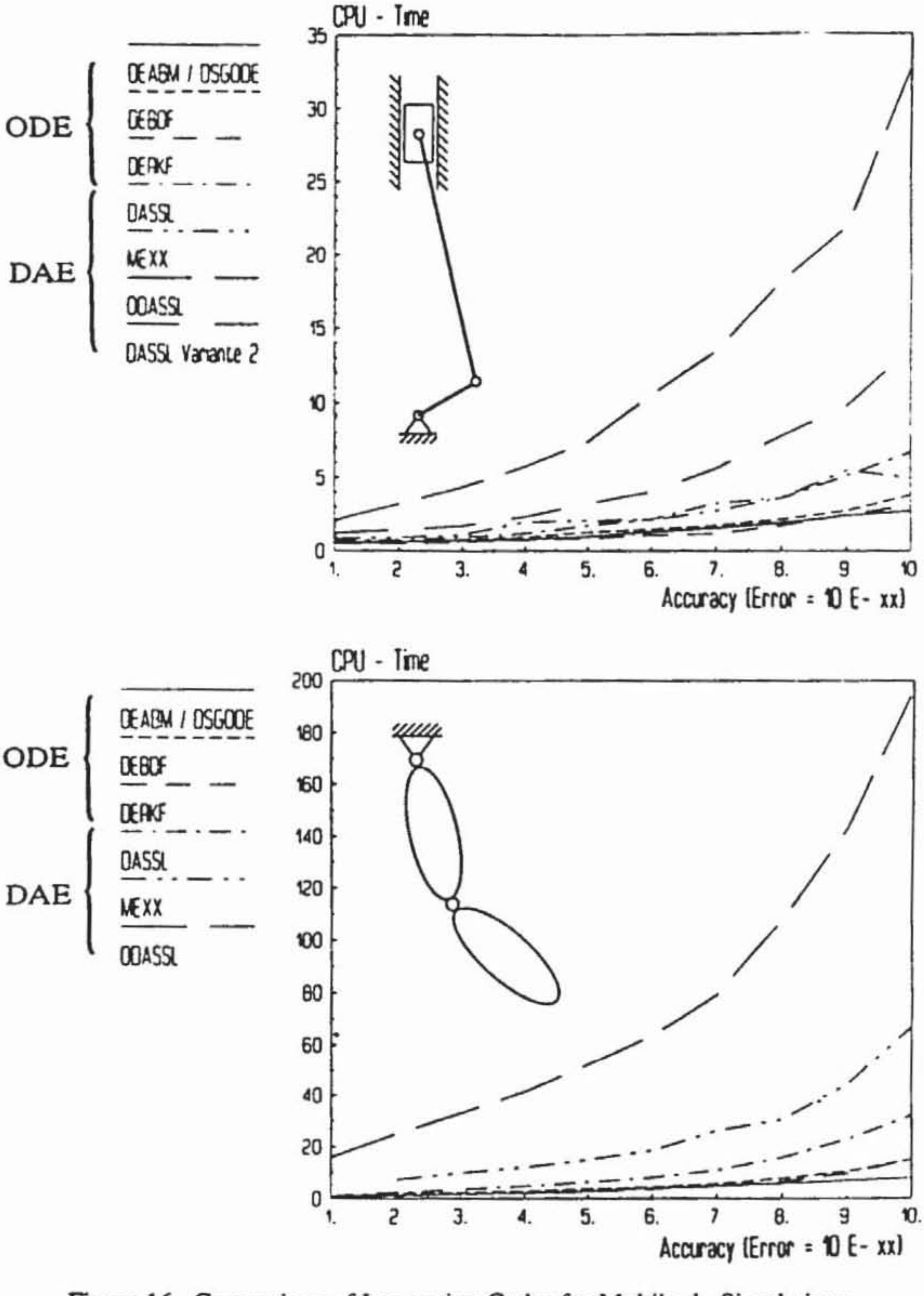
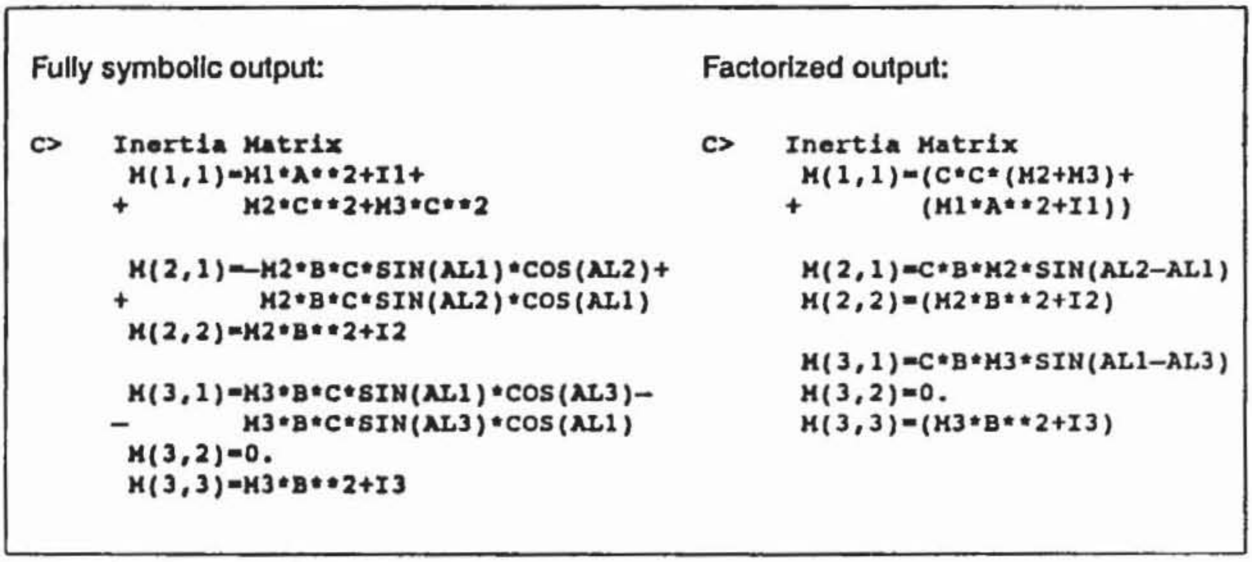


Figure 16. Comparison of Integration Codes for Multibody Simulations

Control parameters for compression and factorization enable the user to change the structure of the output equations, Figure 17. For example, the user may want to obtain fully symbolic equations of motion in order to check the results for modeling and input errors. Later, computationally efficient compressed equations can be generated for the verified model.



Fully symbolic output:

```
C>    Inertia Matrix
       M(1,1)=M1*A**2+I1+
      +        M2*C**2+M3*C**2

       M(2,1)=-M2*B*C*SIN(AL1)*COS(AL2)+
      +         M2*B*C*SIN(AL2)*COS(AL1)
       M(2,2)=M2*B**2+I2

       M(3,1)=M3*B*C*SIN(AL1)*COS(AL3)-
      -        M3*B*C*SIN(AL3)*COS(AL1)
       M(3,2)=0.
       M(3,3)=M3*B**2+I3
```

Factorized output:

```
C>    Inertia Matrix
       M(1,1)=(C*C*(M2+M3)+
      +          (M1*A**2+I1))

       M(2,1)=C*B*M2*SIN(AL2-AL1)
       M(2,2)=(M2*B**2+I2)

       M(3,1)=C*B*M3*SIN(AL1-AL3)
       M(3,2)=0.
       M(3,3)=(M3*B**2+I3)
```

Figure 17. NEWEUL Output of Fully Symbolic and Factorized Inertia Matrix

The software module NEWSIM allows the simulation of the symbolic equations of motion provided by module NEWEUL. It automatically generates a problem specific simulation program. The user simply has to add the specification of

- force laws,
- system parameter values, and
- initial conditions.

The simulation results are stored in ASCII data files that can be visualized with arbitrary graphics packages. The software structure is shown in Figure 18.

The simulation results may contain

- the time history of the state variables,
- the kinematical data of observation points,
- data for animation,
- the time history of the reaction forces, and
- user-defined output data.

Apart from time simulations, additional analyses can be performed with the module NEWSIM. These additional features include:

- the quasi static analysis,
- the computation of the state of equilibrium and
- the treatment of the inverse dynamics problem.

The software package NEWEUL is written in FORTRAN 77 and can be implemented on any workstation or mainframe with a FORTRAN 77 compiler. NEWEUL uses its own formula manipulator.

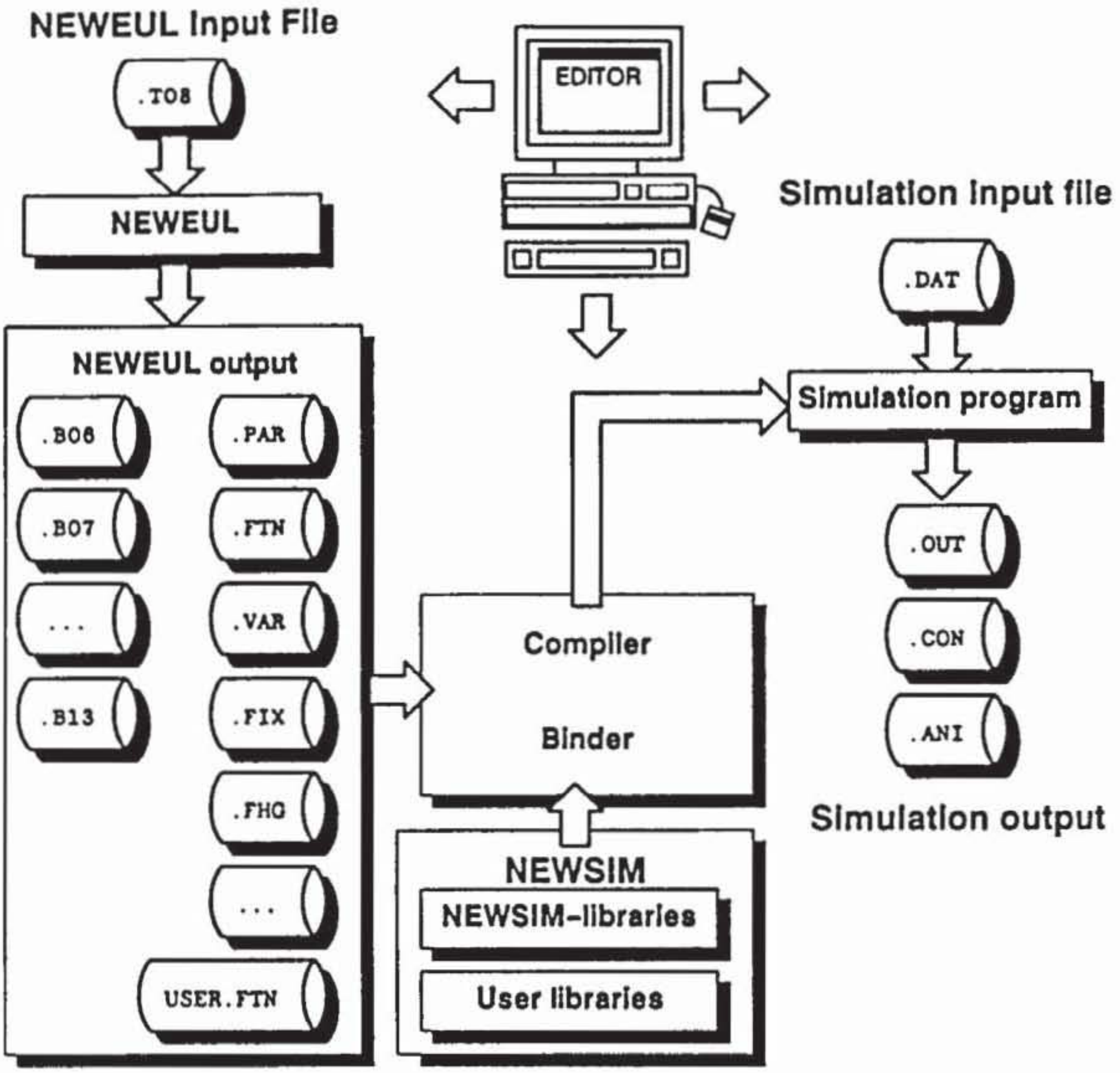


Figure 18. Software Structure of NEWEUL

5 Animation and Human Perception

The simulation produces a large number of data which can be stored in the database. But it is necessary to extract the most valuable information from the simulation data. This can be done by visualization of the motion or by introducing suitable performance criteria, one of them is the human perception of mechanical vibrations.

5.1 Animation of Motion

The visualization of motion can be achieved by the CAD-3D-System or by a separate visualization software tool, respectively, Figure 19. The animation principle is shown in

Figure 20. The information on the 3x1–translation vector $\mathbf{r}_i$ and the 3x3–rotation tensor S_i follows according to (7) from the simulation. This information is used in the 4x4–Denavit-Hardenberg matrix for instancing of the geometric structure at each time step. Daberkow [14] used in the visualization software tool VISANI geometric modeling by polygon sets, as shown in Figure 21, and the PHIGS standard.

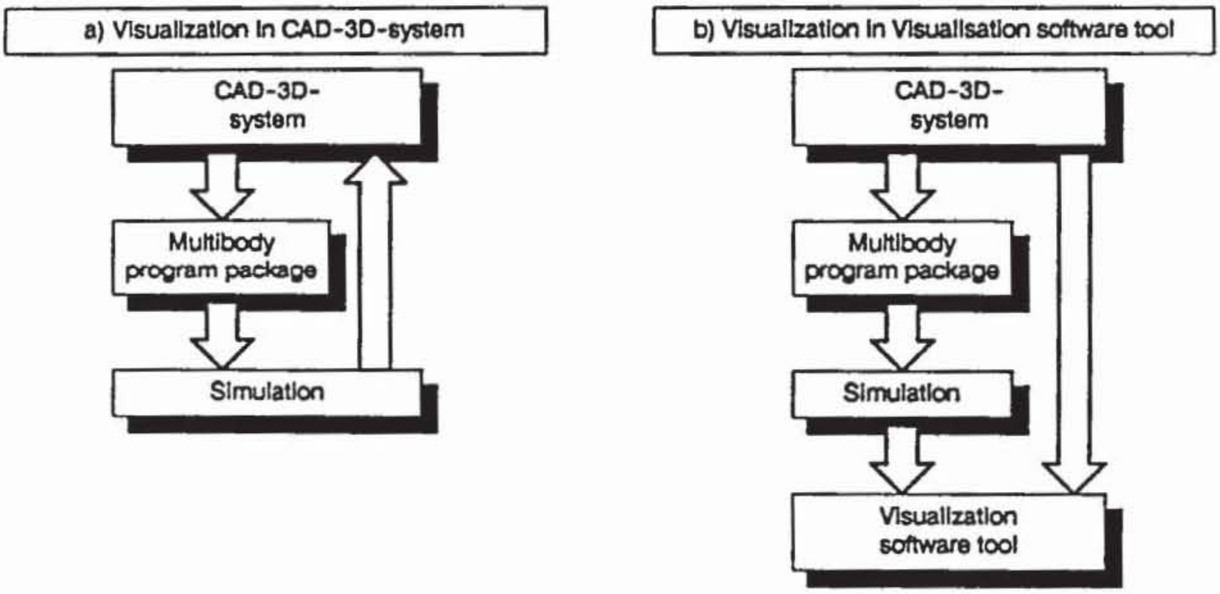

a) Visualization in CAD-3D-system
CAD-3D-system
Multibody program package
Simulation
b) Visualization in Visualisation software tool
CAD-3D-system
Multibody program package
Simulation
Visualization software tool

Figure 19. Visualization of Motion

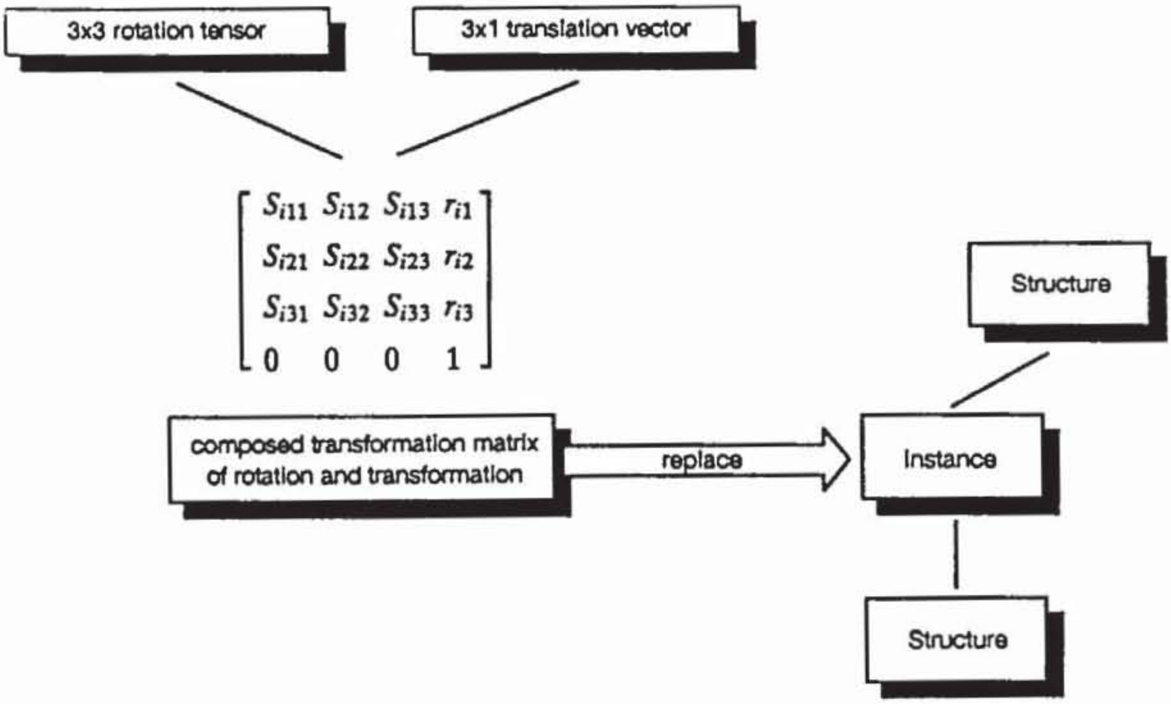
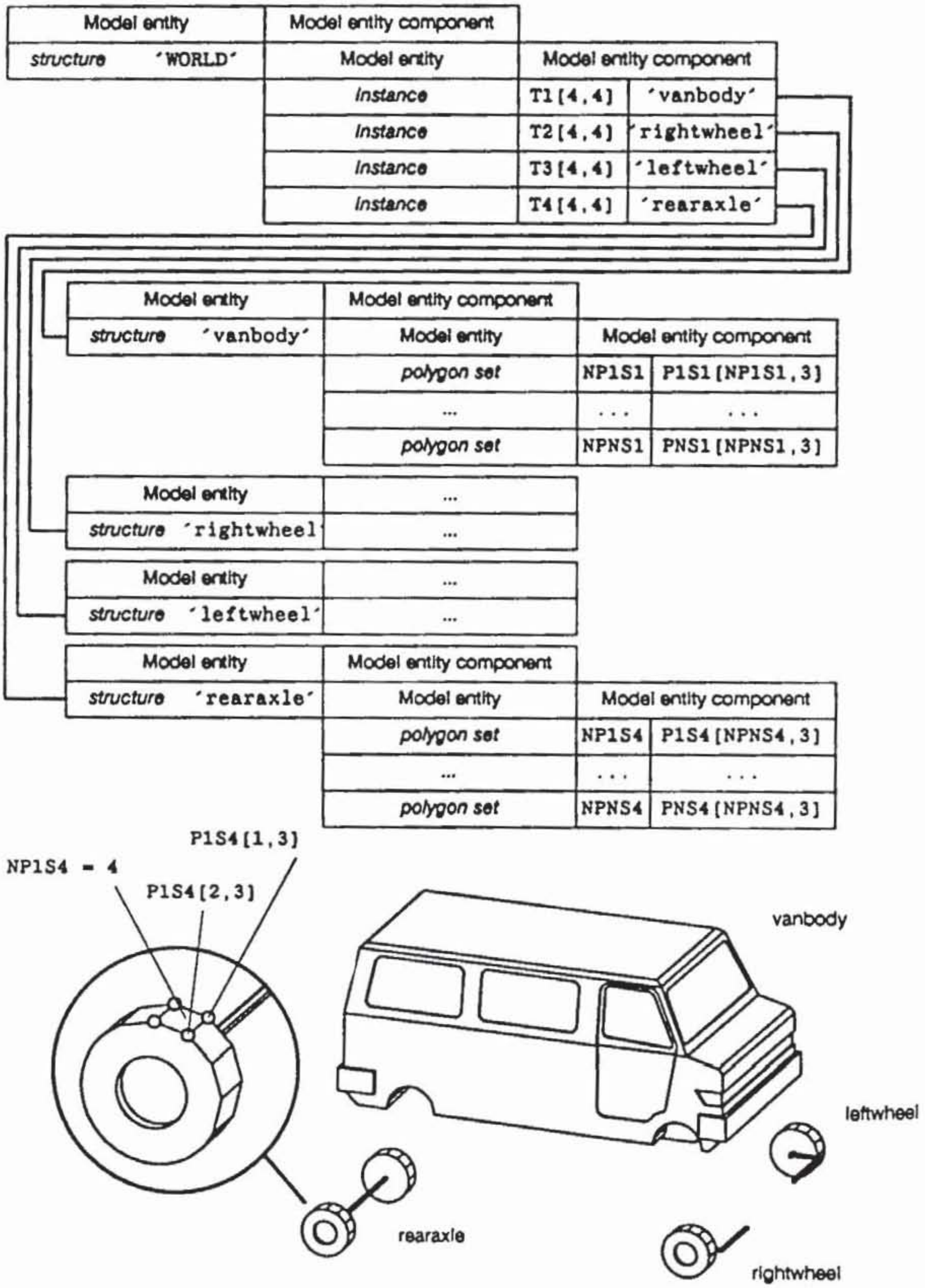

3x3 rotation tensor
3x1 translation vector
S_{i11} S_{i12} S_{i13} r_{i1}
S_{i21} S_{i22} S_{i23} r_{i2}
S_{i31} S_{i32} S_{i33} r_{i3}
0 0 0 1
composed transformation matrix of rotation and transformation
replace
Structure
Instance
Structure

Figure 20. Animation Principle



Model entity		Model entity component		
structure	'WORLD'	Model entity	Model entity component	
		instance	T1[4,4]	'vanbody'
		instance	T2[4,4]	'rightwheel'
		instance	T3[4,4]	'leftwheel'
		instance	T4[4,4]	'rearaxle'

Model entity		Model entity component		
structure	'vanbody'	Model entity	Model entity component	
		polygon set	NP1S1	P1S1[NP1S1,3]
		...	. . .	. . .
		polygon set	NPNS1	PNS1[NPNS1,3]

Model entity		...
structure	'rightwheel	...

Model entity		...
structure	'leftwheel'	...

Model entity		Model entity component		
structure	'rearaxle'	Model entity	Model entity component	
		polygon set	NP1S4	P1S4[NPNS4,3]
		...	. . .	. . .
		polygon set	NPNS4	PNS4[NPNS4,3]

Figure 21. Visualization Software Tool

An overtaking maneuver of a van is presented in Figure 22. The gross motion can be easily observed. However, information on the vertical acceleration acting on the driver is not visible.

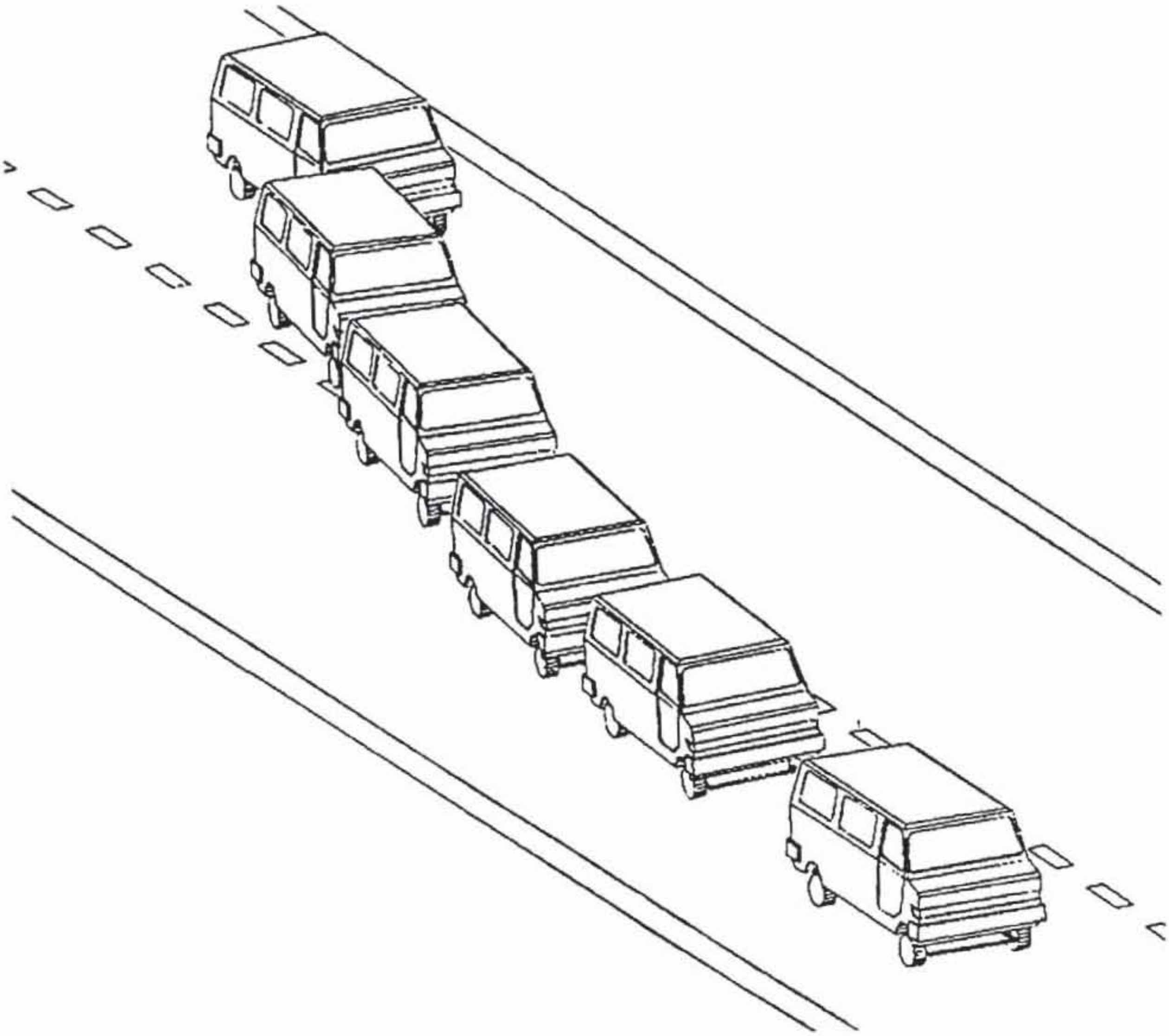


Figure 22. Overtaking Maneuver of a Van

5.2 Human Perception

Even if the road roughness is small and cannot be visualized in an animation, the roughness generates mechanical vibrations of the vehicle acting on the human body. These vibrations depend on the vehicle model, its equations of motion, and the excitation by the road. Since the road roughness represents a stochastic process, the resulting vehicle vibrations are random, too. Therefore, stochastic methods of signal analysis have to be used.

With respect to vehicle vibrations, only the scalar vertical acceleration a(t) will be considered as a criteria for the riding comfort. Then, the perception follows as

$$K = \sigma_{\bar{a}}^2 = \alpha^2 \int_{-\infty}^{+\infty} \left| f_a(\omega) \right|^2 S_a(\omega)\, d\omega \tag{39}$$

where $\alpha = 20\ s^2/m$ is a constant, $f_a(\omega)$ is the frequency response of the vertical sensation and $S_a(\omega)$ is the spectral density of the vertical acceleration a(t). The frequency response $f_a(\omega)$ is given by the international standard [28] and shown in Figure 23. In the time domain, the frequency response is replaced by a second order shape filter

$$\bar{\mathbf{a}}(t) = \overline{\mathbf{h}}^T \bar{\mathbf{v}}(t), \qquad \dot{\bar{\mathbf{v}}}(t) = \overline{\mathbf{F}}\,\bar{\mathbf{v}}(t) + \bar{\mathbf{g}}\,a(t), \tag{40}$$

where $\bar{\mathbf{v}}(t)$ is the 2xl–filter state vector, $\overline{\mathbf{F}}$ the 2x2–filter matrix, $\bar{\mathbf{g}}$ the 2xl–input vector and $\overline{\mathbf{h}}$ the 2xl–output vector. A frequency response of the filter (40) is shown in Figure 23, too. Now the perception reads as

$$K = \sigma_{\bar{a}}^2 = \alpha^2\, \overline{\mathbf{h}}^T\, \mathbf{P}_{\bar{v}}\, \overline{\mathbf{h}} \tag{41}$$

where $\mathbf{P}_{\bar{v}}$ is the 2x2–covariance matrix of the shape filter process $\bar{\mathbf{v}}(t)$.

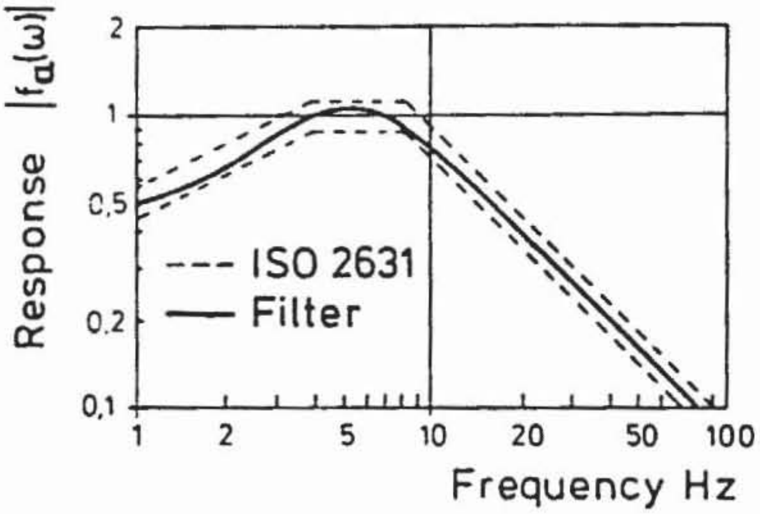


Figure 23. Frequency Response of Human Sensation

The final results (39) and (41) are given in the frequency domain and the time domain, respectively. It turns out that an infinite integral has to be evaluated in the frequency domain while in the time domain, only an algebraic matrix operation is required. Therefore, the covariance analysis using the time domain is preferable [29].

In Figure 24, a complex vehicle is shown consisting of 4 mass points and 7 rigid bodies subject to 35 constraints resulting in f=l9 degrees of freedom. In addition, there are two serial spring–damper–configurations at the engine. Further, four first order excitation shape filters are considered while the sensation shape filter will be neglected. Then, the global system has the order n=44. The human sensation of mechanical vibration will be discussed only with respect to the vertical acceleration of the car body. Numerical results for this complex vehicle have been published by Kreuzer and Rill [30]. Figure 25 shows the RMS value or standard deviation, respectively, of the vertical acceleration normalized by the earth acceleration for each location (C, D) on the car body. It turns out, that the optimal ride characteristics are found in the middle of the car body. Thus, qualitative experience and quantitative measurements are confirmed by theoretical system analysis very well.

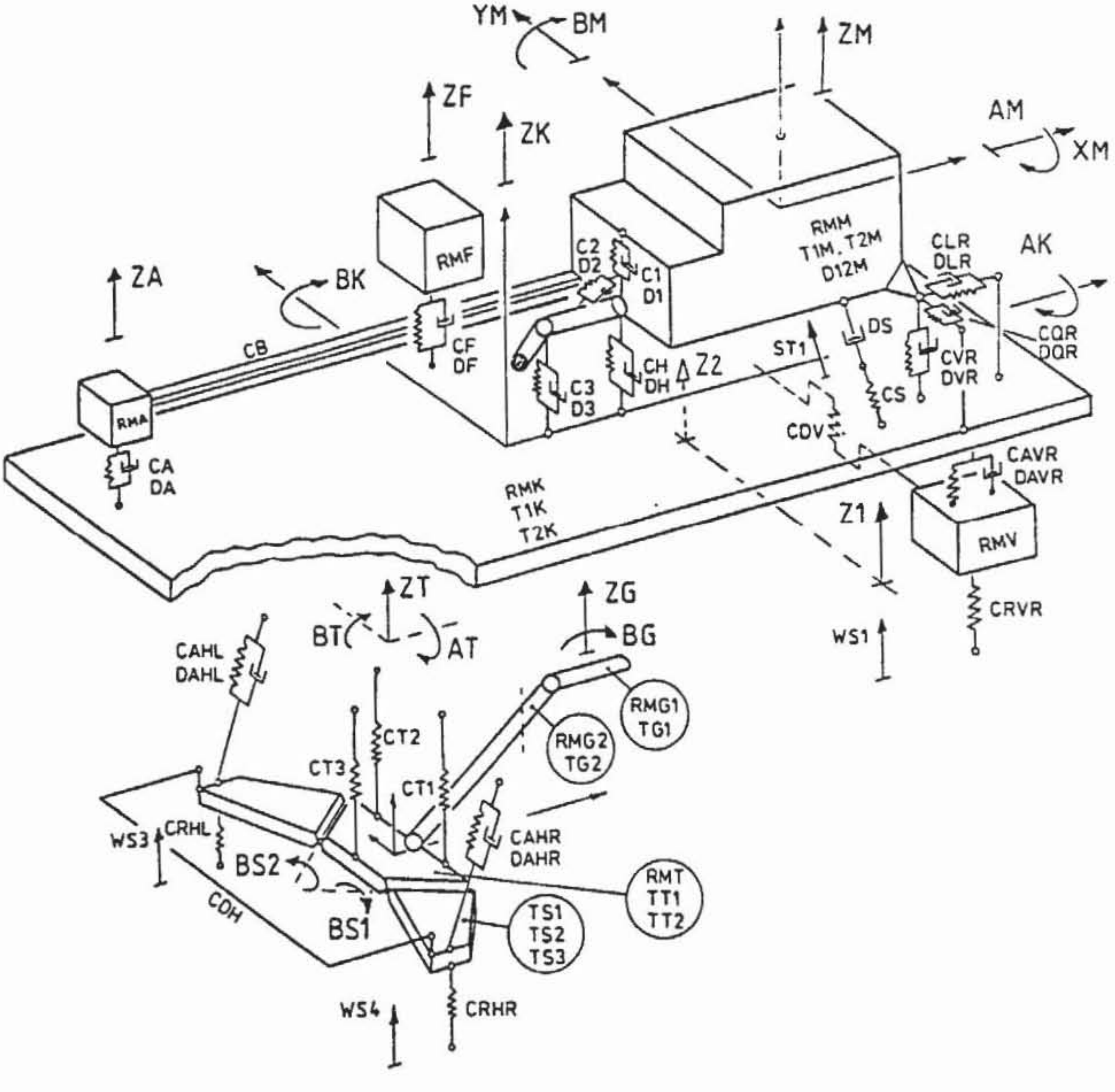


Figure 24. Model of Complex Vehicle under Random Excitation

6 Conclusion

A simulation based design of automotive systems requires modeling using CAD data and well defined multibody system datamodels. The object-oriented approach of modern software engineering is most adequate for multibody system data. Formalisms generating symbolic equations of motion are efficient for dynamical analysis, simulation, and optimization of

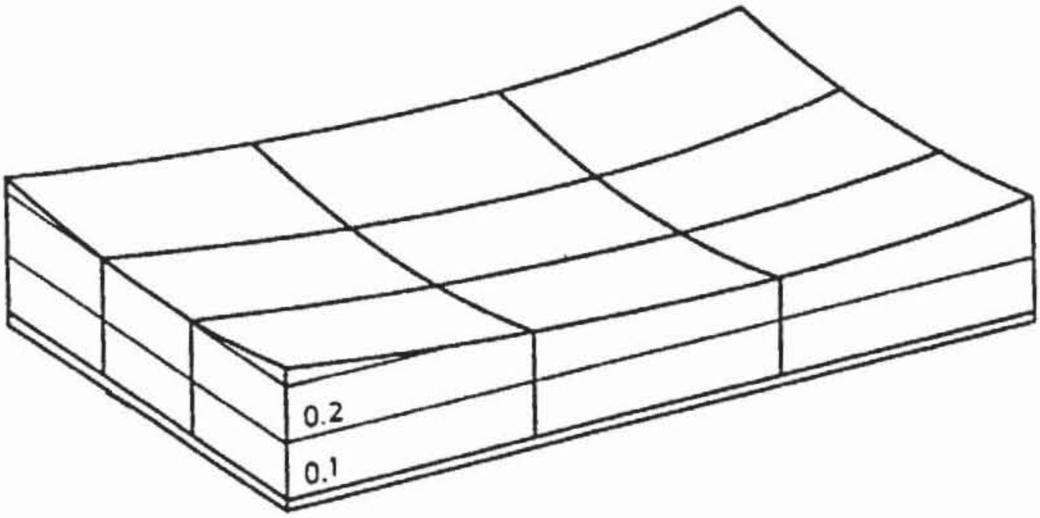


Figure 25. Vehicle Standard Deviation of Acceleration Vehicle Body related to Earth Acceleration

vehicle systems. Numerical simulation codes for the ordinary differential equation representations are superior to differential algebraical equations representations as shown with two examples. The evaluation of simulation data cannot be restricted to graphical interpretation by animation. Advanced vehicle design has to consider quite a number of performance criteria and requirements, e.g., the vertical frequency-weighted random accelerations. Software engineering concepts are most important to vehicle analysis tasks within the overall Concurrent Engineering procedure. A method for the optimization of automotive systems is presented by Bestle [31] in this volume, too.

References

1. Wittenburg, J.: Dynamics of systems of rigid bodies. Stuttgart: Teubner 1977.
2. Schiehlen, W.: Technische Dynamik. Stuttgart: Teubner 1986.
3. Roberson, R.E. and Schwertassek, R.: Dynamics of multibody systems. Berlin, Springer Verlag 1988.
4. Nikravesh, P.E.: Computer-aided analysis of mechanical systems. New Jersey: Prentice-Hall 1988.
5. Haug, E.J.: Computer aided kinematics and dynamics of mechanical systems. Boston: Allyn and Bacon 1989.
6. Shabana, A.: Dynamics of multibody systems. New York: Wiley 1989.
7. Magnus, K (ed.): Dynamics of multibody systems. Berlin: Springer-Verlag 1978
8. Slibar, A; Springer, H. (eds.): Dynamics of vehicles on roads and railway tracks, Swets and Zeitlinger 1978.
9. Haug, E.J. (ed.): Computer aided analysis and optimization of mechanical system dynamics. Berlin: Springer-Verlag 1984.
10. Kortüm, W.; Schiehlen, W.: General purpose vehicle system dynamics software based on multibody formalisms. Vehicle System Dynamics 14(1985), pp. 229-263.
11. Bianchi, G.; Schiehlen, W. (eds.): Dynamics of multibody systems. Berlin: Springer-Verlag 1986.
12. Kortüm, W.; Sharp, R.S.: A report on the state-of-affairs on "Application of multibody computer codes to vehicle system dynamics". Vehicle System Dynamics 20(1991), pp. 177-184.
13. Schiehlen, W. (ed.): Multibody systems handbook. Berlin: Springer-Verlag 1990.
14. Daberkow, A.: Zur CAD-gestützten Modellierung von Mehrkörpersystemen. Ph.D. Thesis. Stuttgart: University of Stuttgart, to appear.

15. Schiehlen, W.: Prospects of the German multibody system research project on vehicle dynamics simulation. In: Dynamics of Vehicles on Road and Tracks. Proc. 12th IAVSD Symposium. Amsterdam: Swet and Zeitlinger 1992, pp. 537-550.
16. Otter, M., Hocke, M., Daberkow, A., Leister, G.: Ein objektorientiertes Datenmodell zur Beschreibung von Mehrkörpersystemen unter Verwendung von RSYST. Institutsbericht IB-16. Stuttgart, Institut B für Mechanik, 1990.
17. Lang, U.: Erstellen von Anwendungsmoduln in RSYST. Stuttgart, Rechenzentrum der Universität Stuttgart, 1988.
18. Loebich, I.: Einführung in RSYST. Stuttgart, Rechenzentrum der Universität Stuttgart, 1988.
19. Rühle, R. et al.: RSYST Unterprogramm– und Moduldokumentation, Version 3.5.0. Stuttgart, Rechenzentrum der Universität Stuttgart, 1988.
20. Schiehlen, W.: Modeling of complex vehicle systems. In: Proc. 8th IAVSD Symposium. Hedrich, J.K. (ed.) Lisse: Swets & Zeitlinger 1984, pp. 548-563.
21. Hollerbach, J.M.: A recursive Lagrangian formulation of manipulator dynamics and a comparative study of dynamics formulation complexity. IEEE Trans. Sys. Man. Cyb. 10(1983), 730-736.
22. Walker, M.W.; Orin, D.E.: Efficient dynamic computer simulation of robot mechanisms. J. Dyn. Sys. Meas., Control 104(1982), 205-211.
23. Brandl, H.; Johanni, R.; Otter, M.: A very efficient algorithm for the simulation of robots and similar systems without inversion of the mass matrix. In: Theory of Robots. Kopacek, P.; Troch, I.; Desoyer, K. (eds.): Oxford: Pergamon Press 1988, pp. 95-100.
24. Schiehlen, W.: Computational aspects in multibody system dynamics. Comp. Meth. Appl. Mech. Eng. 90(1991), pp. 569-582.
25. Leister, G.: Wahl geeigneter Koordinaten zur Dynamikanalyse von Mehrkörpersystemen. Zwischenbericht ZB-49 Stuttgart: Institute B of Mechanik 1990.
26. Valasek, M.: On the efficient implementation of multibody systems formalisms. Institutsbericht IB-17. Stuttgart Institute B of Mechanics 1990.
27. Leister, G.: Beschreibung und Simulation von Mehrkörpersystemen mit geschlossenen kinematischen Schleifen. Fortschr. Ber. VDI Reihe 11 Nr. 169. Düsseldorf: VDI-Verlag 1992.
28. International Standard ISO 2631, Guide for the evaluation of human exposure to whole-body vibrations. Int. Org. Standardization (1974).
29. Schiehlen, W.: Vehicle system dynamics. In: Theoretical and Applied Mechanics. Niordson, F.; Olhoff, N. (eds.): Amsterdam: North-Holland 1985, pp. 387-398.
30. Kreuzer, E.; Rill, G.: Vergleichende Untersuchung von Fahrzeugschwingungen an räumlichen Ersatzmodellen. Ing. Arch. 52(1982), pp. 205-219.
31. Bestle, D.: Optimization of automotive systems. (in this volume)