

SYMBOLIC COMPUTATIONS IN MULTIBODY SYSTEMS

W. SCHIEHLEN

*Institute B of Mechanics, University of Stuttgart
D-70550 Stuttgart, Germany*

ABSTRACT: Symbolic formula manipulation has proven to be an efficient tool in the dynamical analysis of multibody systems. A multibody system data base is introduced and its implementation using a CAD-3D-software is shown. Starting from the data base the equations of motion are generated by a coordinate partitioning approach combined with the projection criterion. For the symbolical-numerical solution inverse kinematics algorithms are applied. The simulation results are visualized by computer animation. A four-bar mechanism and a crank-slider mechanism serve as examples. Two approaches for the dynamical analysis of flexible multibody systems are presented. Further, the optimization of multibody systems is treated using an actively controlled vehicle suspension as an example.

1. Introduction

An integrated approach for modeling, generation of symbolical equations of motion, simulation and visualization of multibody systems is described. A general object-oriented data model for all multibody formalisms is presented. With respect to existing CAD-interfaces, different solid model design methods and various visualization demands, the data model allows multibody modeling with a direct interface to a data base. Some software tools like an integrated Newton-Euler formalism are able to use immediately the parametrized multibody system data base. For multibody systems with closed kinematic loops a set of ordinary differential equations is formulated automatically which can be solved with explicit multistep integration algorithms. This is achieved by different minimal sets of generalized coordinates being specified by a coordinate partitioning approach during the numerical integration. The basic steps and the extreme flexibility of this automated mechanical design and simulation process is demonstrated for mechanisms.

Machines, mechanisms, road vehicles and spacecrafts can be modeled properly as multibody systems for the design and the dynamical analysis. 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 e.g. by Roberson and Schwertassek [1], Nikravesh [2], Haug [3] and Shabana [4]. The state-of-the-art is also presented at a series of IUTAM/IAVSD symposia, documented in the corresponding proceedings, see, e.g. Kortüm and Schiehlen [5], Bianchi and Schiehlen [6], Kortüm and Sharp [7].

In addition, a number of commercially distributed computer codes was developed, a summary of which is given in the Multibody System Handbook [8]. 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, 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.

2. Multibody systems data model

Modeling of a mechanical system by the method of multibody systems is characterized by a composition of rigid bodies, joints, springs, dampers, and servomotors, see Figure 1. Force elements like springs, dampers, and servomotors acting in discrete nodal points result in applied forces and torques on the rigid bodies. Joints with different properties connecting the various bodies constrain their motion, they are often identified as constraint elements.

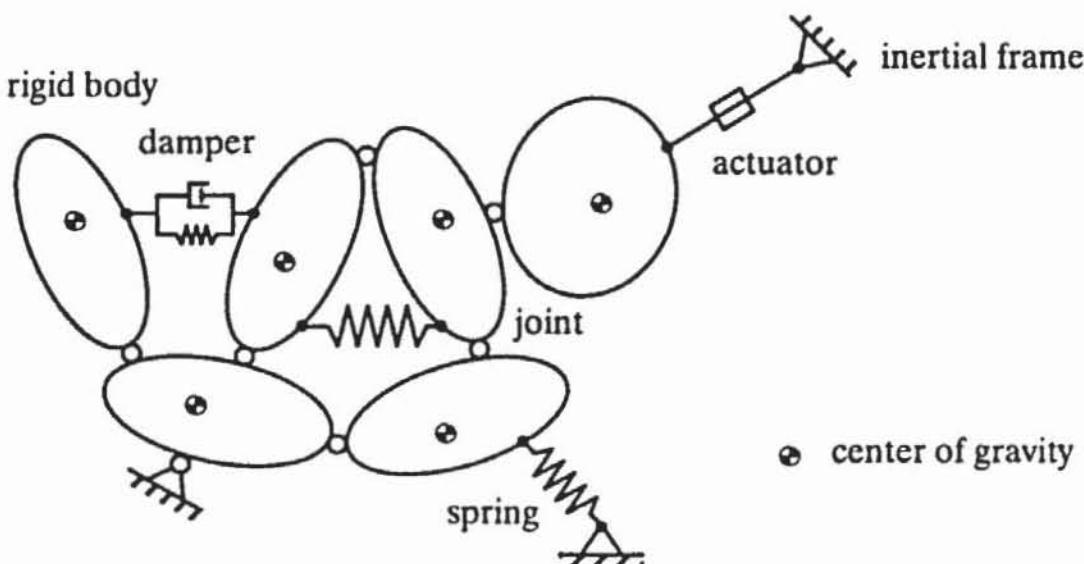


Figure 1: Multibody System

For the generation of the equations of motion computer programs may be used. Well known multibody system computer codes producing exclusively numerical data are ADAMS, Orlandea [9], and DADS, Haug [10]. To the contrary, computer programs like SD-FAST, Rosenthal and Sherman [11] and NEWEUL, Kreuzer [12] provide the explicit symbolical expressions for the system equations.

Nowadays CAD-systems are widely embedded in the industrial design and construction process, while a general application of three-dimensional CAD-systems is still rare. They support an analytically and topologically complete modeling, a collision detection, and the calculation of surface and volume properties closely related to the geometric representation of solid models.

Some couplings of solid modelers with multibody simulation software are realized for the numerical computer code ADAMS, e.g. for the CAD-system ARIES [13]. A CAD-3D-system independent approach is included in the program package RASNA and is described by Hollar and Rosenthal [14].

A system dynamics analysis requires as basic parameters mass, center of gravity, and moments of inertia of each body related to the geometry model and modeling method of the CAD-system used. A modular software concept demands an exchange of complete or single object data between the CAD-system and the multibody formalism. Therefore, a general interface to multibody computer codes is demanded to serve as a compatible and comfortable CAD-post processor, taking the different algorithms and implementations of multibody computer codes into account. The commercially available multibody modeling software tools within CAD-systems are mostly dedicated to a particular multibody dynamics computer code. Often, no options are supplied for a parametric multibody system description or the modeling is restricted to either robot, mechanism or vehicle dynamics. This variety of systems, each with different model data and the growing problems in the exchange of data, requires the development and production of cheaper and more reliable software products.

Consequently this leads to a database concept for the CAD-3D-modeling of multibody systems, see Figure 2:

- Collect the necessary data describing uniquely a multibody model for the different multibody programs.
- Examine the different geometry models of CAD-systems for solids and extract the relevant data for multibody systems.
- Define a geometry model for the representation of multibody elements.
- Design data types and operations and construct a software interface for a code-independent modeling of multibody systems.

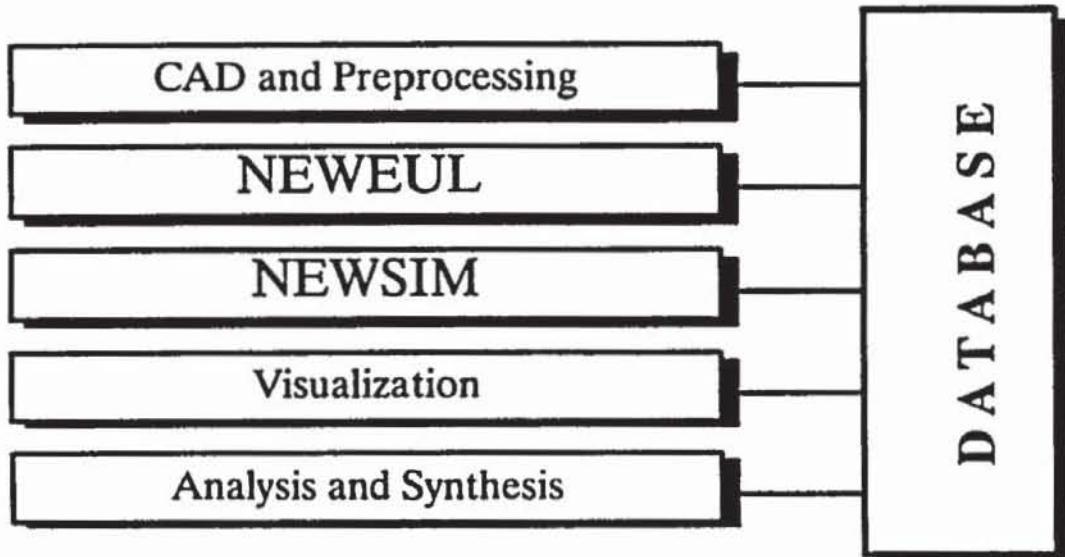


Figure 2: Modules within the database concept

A dynamic simulation environment for multibody systems represents in practice a large, sophisticated software system. Therefore, an important step is the definition of an abstract data model on a conceptual level. A first effort to develop a generalized data model for multibody systems including symbolical parameters and a postprocessing of CAD-data is described by Otter, Hocke, Daberkow, and Leister [15]. Each of the bodies is described by body-fixed reference frames. Further body-fixed frames, related joints and force elements are described. Additional symbolical parameters are defined for the position and orientation of the frames with respect to each other as well as the mass properties of the bodies. Consequently, for symbolical as well as numerical formalisms a generalized data base relies upon the basic modeling elements frame, body, joint, and force and is further adapted and extended with respect to the geometry models in CAD-3D and graphics systems.

A property of a solid in a CAD-3D system can be derived from a face normal specifying the inner and outer parts of an object, while the coincidence of the vertices of adjoining faces is not guaranteed. The geometric modeling by parametrized shapes is appropriate for geometric objects, whose shape is uniquely defined by a restricted number of parameters. Examples of parametrized shapes with an equivalent wire representation are shown in Figure 3.

For the global properties volume, surface area, moment of inertia, and center of gravity of solid models, integrals have to be evaluated like

$$I = \int_{Solid} f^V dV \quad (1)$$

see e.g. Mortenson [16], where $f^V = f^V(x, y, z)$ denotes a scalar property function.

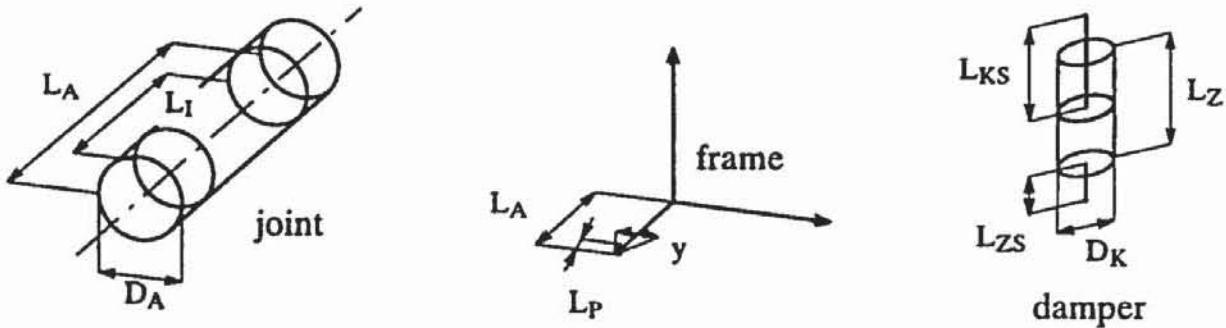


Figure 3: Parametrized wire representations of multibody elements

While Constructive Solid Geometry suggests the calculation of mass properties by the following recursively applied formulas

$$\begin{aligned} \int_{Solid1 \cup Solid2} f^V dV &= \int_{Solid1} f^V dV + \int_{Solid2} f^V dV - \int_{Solid1 \cap Solid2} f^V dV, \\ \int_{Solid1 - Solid2} f^V dV &= \int_{Solid1} f^V dV - \int_{Solid1 \cap Solid2} f^V dV, \end{aligned} \quad (2)$$

boundary representations allow the evaluation via surface integrals.

The examination of different geometry models yield the following results:

- Mass property calculation modules for multibody systems do not depend on the model geometry (CSG or B-Rep). These results can be related directly with the input entities needed for the rigid bodies.
- A planar face model derived from the geometric entities of the solid body yield the graphic data for the description of the body's shape necessary for visualization.
- The parametrized shapes are well suited to serve as a geometry model for multibody modeling elements like frame, joint, and force.

The object-oriented data model conceptually developed by Otter et al. [15] results in classes defined for the elements *part*, *frame*, *body*, *interact*, *joint*, *force*, *global*, and *param* and additional operations valid for these classes.

An object of class *body*, e.g. Figure 4, comprises alle time-invariant data of a rigid body. It is obvious that the components inertia matrix and mass of an object of class *body* are supplied by their numerical values, too. A location of the center of gravity different from the body-fixed reference frame is taken into consideration by reference to an equivalent object of class *frame*.

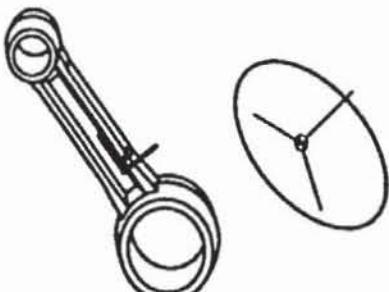
Technical system	multibody model	class description	instance of class																																		
		<table border="1"> <thead> <tr> <th colspan="3">class: body</th> </tr> <tr> <th>name</th> <th>type</th> <th>description</th> </tr> </thead> <tbody> <tr> <td>mass</td> <td>dparam</td> <td>mass of body</td> </tr> <tr> <td>mframe</td> <td>name</td> <td>c.o.g frame</td> </tr> <tr> <td>inertia</td> <td>dparam(6)</td> <td>inertia tensor</td> </tr> <tr> <td>iframe</td> <td>name</td> <td>tensor frame</td> </tr> </tbody> </table>	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	<table border="1"> <thead> <tr> <th colspan="2">object: body1</th> </tr> <tr> <th>name</th> <th>component</th> </tr> </thead> <tbody> <tr> <td>mass</td> <td>M1</td> </tr> <tr> <td>mass</td> <td>0.418</td> </tr> <tr> <td>mframe</td> <td>''</td> </tr> <tr> <td>inertia</td> <td>I11,I22,I33,I12,I13,I23</td> </tr> <tr> <td>inertia</td> <td>113.0,1296.49,1380.67</td> </tr> <tr> <td>iframe</td> <td>''</td> </tr> </tbody> </table>	object: body1		name	component	mass	M1	mass	0.418	mframe	''	inertia	I11,I22,I33,I12,I13,I23	inertia	113.0,1296.49,1380.67	iframe	''
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																																				
mass	0.418																																				
mframe	''																																				
inertia	I11,I22,I33,I12,I13,I23																																				
inertia	113.0,1296.49,1380.67																																				
iframe	''																																				

Figure 4: Object of class *body* with its data model

Coupling elements of a multibody system are collected in class *interact*. Interactions are valid between two objects of class *frame* on different objects of class *part*.

Due to object-oriented software techniques, the definition of abstract data types in classes furthermore demands a description of the operations valid on the objects. These operations are designed for a practical, interactive multibody modeling process, e.g. in a CAD-3D-system. For all classes the basic operations 'create', 'delete', 'modify', and 'list' are defined, more complex operations take the relationships between objects of a multibody system into account.

Further classes are required for the graphical representation, like the actual frame axis length, its color or visibility, which depend on the actual multibody size and modeling state. An equivalent geometry data model for multibody elements well suitable for machine, robot and vehicle dynamics requires a unique spatial representation of the multibody elements, their function and physical quantity, see Daberkow [17]. From Figure 3 it is obvious that spatial parametrized shapes satisfy a graphic representation for objects of class *frame*, *joint*, and *force*. The definition of the geometry 3D classes *g3frame*, *g3joint* and *g3force* and operations for the geometry data model is equivalent to the multibody data model and includes classes comprising color, projection and viewpoint data.

3. Implementation and CAD-3D-realization

The implementation of the object-oriented data model in the data base system RSYST [18] allows storage and modification of multibody system objects. To realize

fast access and interactive graphic visualization, the implementation of the object-oriented classes and operations within the CAD-3D-system is performed by means of data types and routines, which result in a system-independent modeling kernel library for multibody systems, see Daberkow [17]. This high level library DAMOS-C (DAta MOdel Standard implemented in C) supplies interfaces for modeling, input, and output as well as for the graphic representation. This open interface allows the integration in the commercially available CAD-3D- system SIGRAPH [19] and a new developed graphics-system.

The integration scheme in Figure 5 shows the interfaces to the CAD-3D software modules of SIGRAPH. An extension of the CAD command language supplies additional commands which are necessary for the execution of multibody modeling operations. The CAD-3D-system menu is completed by special multibody system icons. To assure the graphic display of the modeling elements, the parametrized shapes are modeled via the 3D-wireframe entities of the CAD-graphic subsystem. A multibody command language of RSYST serves as a multibody system neutral file to store the multibody objects, see Otter et al. [20]

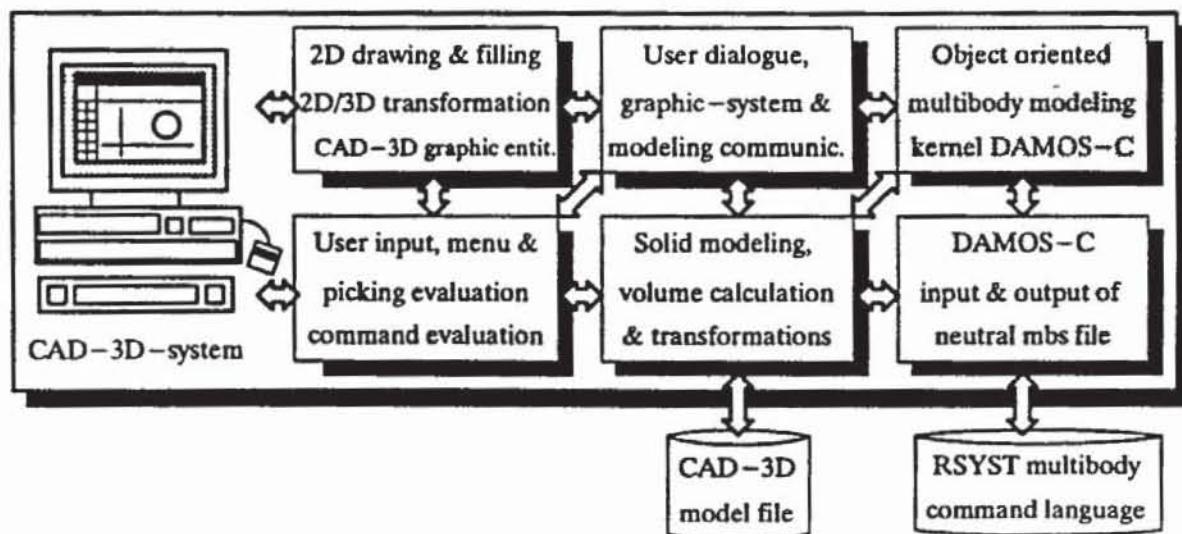


Figure 5: Integration of the multibody modeling kernel

The solid model design of a crank slider mechanism is performed by volume oriented techniques in PARASOLID from a disassembled model, Figure 6. All bodies of the crank-slider mechanism of a single four stroke engine are shown in Figure 6. Each body is supplied with adequate density attributes.

The first multibody modeling step is the initialization. Here, an appropriate solid is chosen as the inertial body of the multibody system, see Figure 6. In the next step arbitrary solids are interactively chosen to have the properties of a multibody part. Each object of class *body* retrieves its mass and inertia components from the

mass property calculation modul of PARASOLID. To visualize the multibody part property, the equivalent solids are supplied by reference frames, located in the center of gravity.

By default, the orientation of further created joint and force definition frames is parallel to the specified reference frame. The position of these frames is defined by the CAD-3D-picking commands performed by the user. Figure 6 shows these modeling steps and the graphic representation of the objects. Joint definition frames are located along the unit normals of those faces, which form bearing surfaces or bearing bores of a solid.

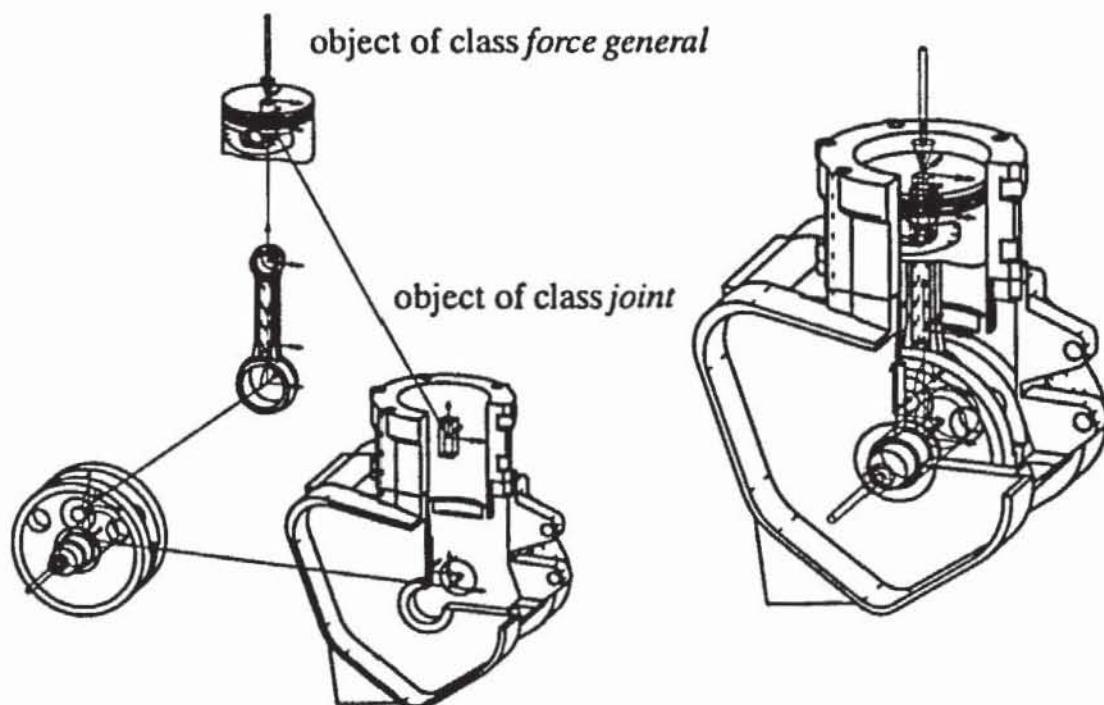


Figure 6: Disassembled and assembled mechanism with joint and force objects

A planar system modeled for spatial analysis demands a proper constraint selection. Redundant constraints remain if a mechanism is supplied with joints of class *revolute* and *translational*, making the determination of reaction forces impossible. Consequently, for an analysis modified joints have to be chosen. Objects of class *revolute* are visualized by the parametrized shapes and the wireframe entities. The connection between the objects of class *part* by the object of class *interact* is visualized by a 3D-line entity between the interacted frames.

The multibody modeling kernel library implemented in the CAD-3D-system supports an assembling of arbitrary pairs of class *part*. Figure 6 shows the assembling of individual solids over the equivalent objects of class *joint*. By modifying the *rangle* component of arbitrary objects of class *joint*, an initial multibody configuration is

adjusted interactively, providing therefore an initial estimate for closed loop systems. Finally, an object of class *force general* is added to the piston part.

4. Generation of equations of motion starting from the database

The generation of equations of motion and the embedding of these equations to simulation software is especially in case of large multibody models very time consuming and prone of errors. Starting from the description of the multibody system stored on the database, the modul NEWEUL, Kreuzer and Leister [21], generates symbolic equations of motions and all information necessary for the automatic simulation. The modul NEWSIM, Leister [22], uses in the next step the compiled symbolical equations of motion for the simulation. Using the object-oriented datamodel the modules NEWEUL and NEWSIM are tools of a modular software package of the multibody system approach, see Figure 7.

In a first step the information stored in the database has to be extracted. In a modular concept the generation of equations of motion and the simulation have to be separated. The datamodel includes all the information necessary for the generation of the equations of motion and, an adapted version of NEWEUL can be used as module in the database concept. Based upon a Newton-Euler formalism the symbolical equations of motion are generated using d'Alembert's or Jourdain's principle to eliminate the reactions forces and torques, see Ref. [23]. By means of a special, for the multibody system approach developed formulamanipulator, it is possible to generate the equations of motion with minimal costs of computation time, see Kreuzer [12]. The symbolical equations of motion can be used on the one hand in the simulation environment NEWSIM and on the other hand in any general purpose simulation environment, e.g. ACSL [24] or DSSIM [25].

At first, from the objects *interact* and *joint* the topology of the multibody system is computed. Additionally from the object *joint* the generalized coordinates are determined. The kinematical description of multibody systems is done by the definition of frames relatively to any arbitrary frame. These frames define rigid bodies, joints, auxiliary frames, and reference frames, too. Additionally the mass-geometric properties and the applied forces and moments are necessary. These data can be found in the objects *interact* and *force*, see Figure 7.

The modul NEWSIM serves for the numerical simulation of the generated symbolic equations of motion. It is easy to study the influence of parameters or to optimize the dynamical behaviour with respect to some specified criteria. NEWSIM has the possibility to treat additional differential or differential-algebraic equations. For

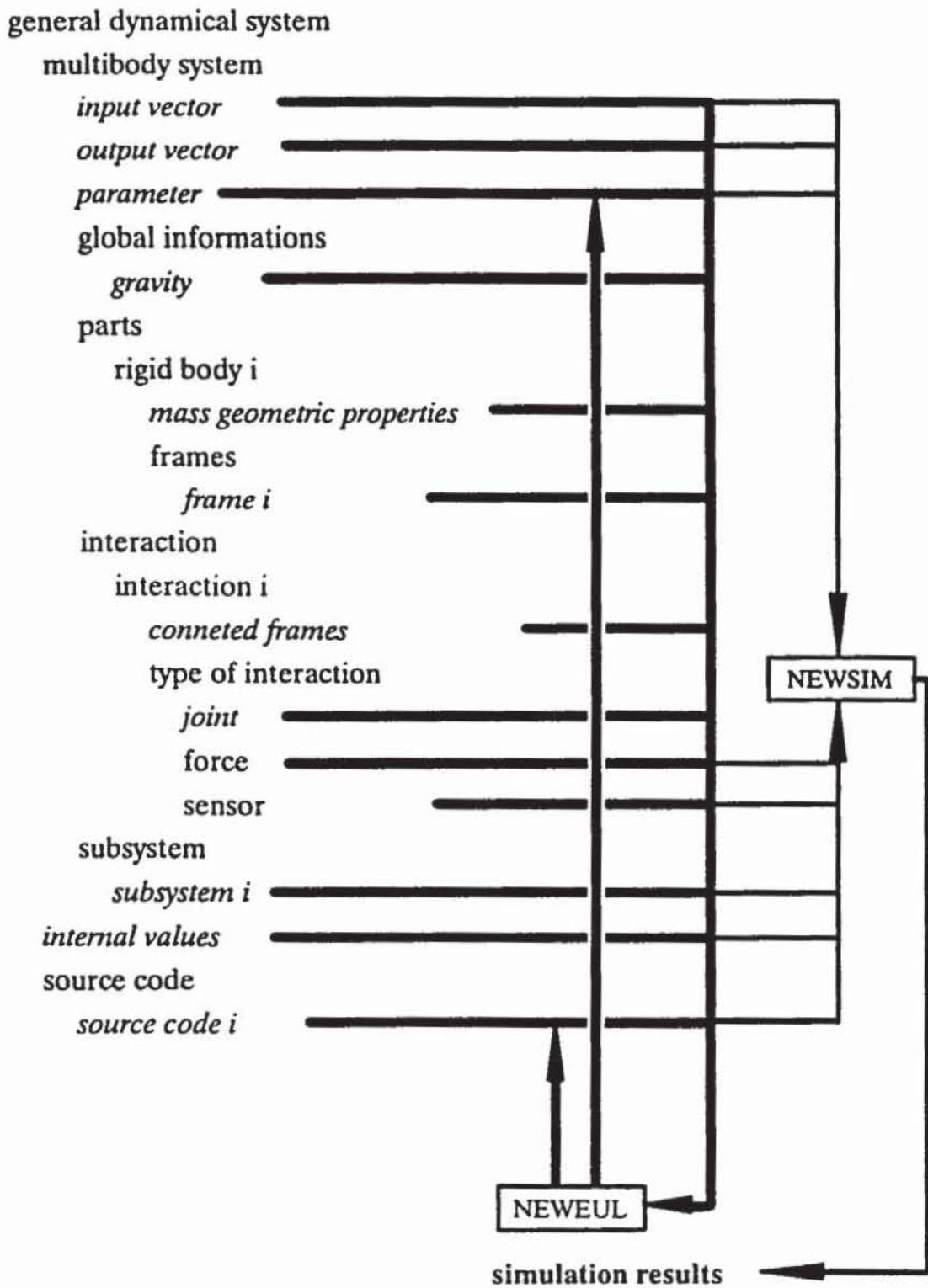


Figure 7: Dataflow of the datamodel

integration in the time-domain different integration schemes are e.g. Runge-Kutta methods, Adams-methods, BDF-methods. For multibody systems including closed loops a modified Adams-Bashforth-Moulton method is implemented, see Leister [26]. All necessary routines for the automatic simulation software are generated by NEWEUL, Figure 8. After the compilation and binding step the problem-specific program takes all parameters and options from the datafile. This program reads all options, initial conditions, fixed system parameters like masses, moments of inertia, geometric data, stiffness constants, and further data from the input file and solves the equations of motion of the problem.

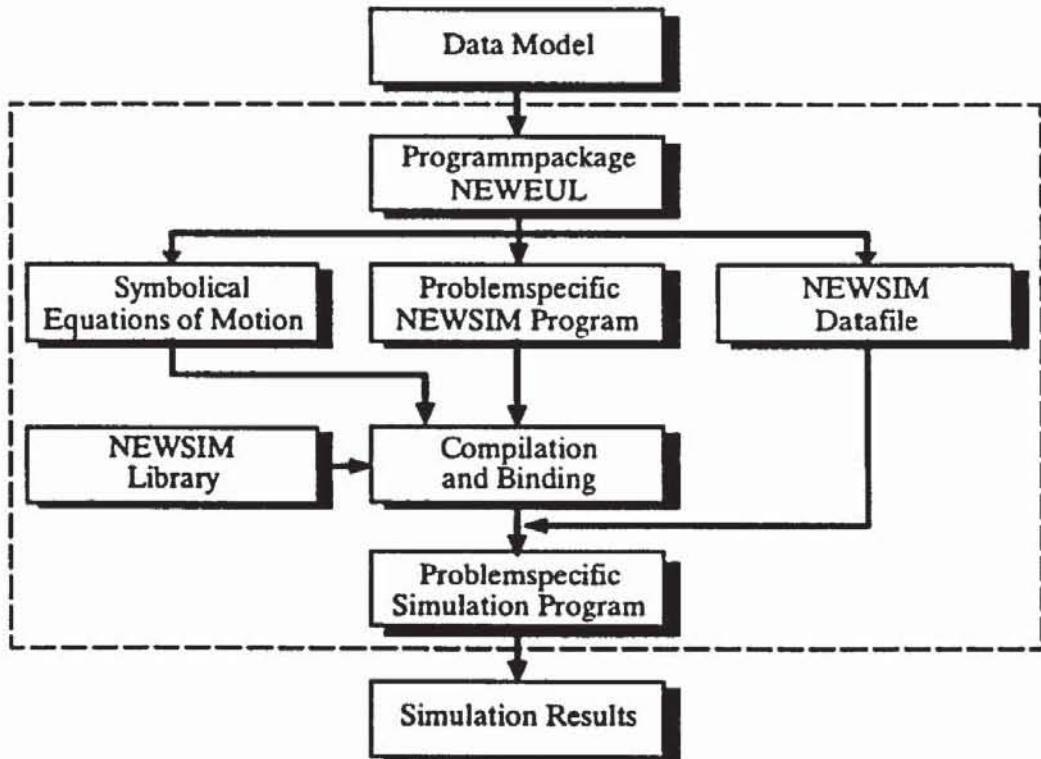


Figure 8: Simulation of the dynamic behaviour with NEWEUL and NEWSIM

5. Formalism for multibody systems using coordinate partitioning

Modeling dynamical systems by the method of multibody systems results in either ordinary differential equations (ODEs) using minimal coordinates or coupled differential and algebraic equations using cartesian and redundant coordinates (DAEs). Often ODEs are integrated numerically by explicit multistep integration algorithms whereas DAEs have to be integrated by implicit or halfimplicit methods. Numerical experiments have shown, Leister [26], that the integration algorithms for ODEs seems to be more efficient than algorithms for DAEs. Thus, it is advantageous to describe

multibody systems by a minimal number of pure differential equations, the so-called state space form.

Consider a mechanical system modelled by e generalized coordinates $\mathbf{x} = [x_1, \dots, x_e]^T$ and subject to q holonomic constraints represented by at least twice differentiable functions $\Phi(\mathbf{x}, t) = [\Phi_1(\mathbf{x}, t), \dots, \Phi_q(\mathbf{x}, t)]^T$. The governing equations of constrained motion of the system can be written in the following DAE form, see e.g. Ref. [23]

$$\mathbf{M}(\mathbf{x}, t) \ddot{\mathbf{x}} = \mathbf{h}(\dot{\mathbf{x}}, \mathbf{x}, t) + \mathbf{Q}(\mathbf{x}, t) \mathbf{g}, \quad (3)$$

$$\Phi(\mathbf{x}, t) = \mathbf{0}, \quad (4)$$

where \mathbf{M} is the $e \times e$ symmetric positive-definite mass matrix; \mathbf{h} represents the components of applied forces on the system and the gyroscopic terms; $\mathbf{Q}^T = \mathbf{C} = \partial\Phi/\partial\mathbf{x}$ is the $q \times e$ constraint matrix; and $\mathbf{g} = [\lambda_1, \dots, \lambda_q]^T$ conserves Lagrange multipliers or generalized constraint forces, respectively. The constraint equations (2) can be differentiated to:

$$\dot{\Phi} = \mathbf{C}(\mathbf{x}, t) \dot{\mathbf{x}} + \mathbf{a}(\mathbf{x}, t) = \mathbf{0}, \quad (5)$$

$$\ddot{\Phi} = \mathbf{C}(\mathbf{x}, t) \ddot{\mathbf{x}} + \mathbf{b}(\dot{\mathbf{x}}, \mathbf{x}, t) = \mathbf{0}, \quad (6)$$

where $\mathbf{a} = \partial\Phi/\partial t$, and $\mathbf{b} = \dot{\mathbf{C}}\dot{\mathbf{x}} + \dot{\mathbf{a}}$.

The coordinate partitioning method makes use of the fact that only $f = e - q$ from the e initial coordinates \mathbf{x} are *independent*, denoted $\mathbf{y} = [y_1, \dots, y_f]^T$; the others are referred to as the *dependent* coordinates in the meaning of this method, $\mathbf{x}_D = [x_{D1}, \dots, x_{Dq}]^T$. Thus, according to the symbolic partition,

$$\mathbf{x} = [\mathbf{y}^T \quad \mathbf{x}_D^T]^T, \quad (7)$$

the constraint equations (5) can be rewritten as

$$\mathbf{C}_I(\mathbf{x}, t) \dot{\mathbf{y}} + \mathbf{C}_D(\mathbf{x}, t) \dot{\mathbf{x}}_D + \mathbf{a}(\mathbf{x}, t) = \mathbf{0}. \quad (8)$$

For clarity in the mathematical formulation, this symbolic notation, partitioned relative to independent and dependent coordinates, will be used through the whole paper. In numerical algorithms, however, it is usually more convenient to complete this task by assigning appropriate *addresses* to the entries of matrices and vectors being partitioned.

If constraints (4) are independent, $\text{rank}(\mathbf{C}) = q$, there exist at least one set of $\dot{\mathbf{x}}_D$ such that the corresponding square submatrix \mathbf{C}_D is nonsingular, $\det(\mathbf{C}_D) \neq 0$.

This enables one to express $\dot{\mathbf{x}}_D$ as linear combinations of $\dot{\mathbf{y}}$, and then $\ddot{\mathbf{x}}_D$ as linear combination of $\ddot{\mathbf{y}}$, i.e.:

$$\dot{\mathbf{x}}_D = -C_D^{-1}(C_I \dot{\mathbf{y}} + \mathbf{a}) = A(\mathbf{x}, t) \dot{\mathbf{y}} + \boldsymbol{\eta}(\mathbf{x}, t), \quad (9)$$

$$\ddot{\mathbf{x}}_D = A(\mathbf{x}, t) \ddot{\mathbf{y}} + \boldsymbol{\xi}(\dot{\mathbf{y}}, \mathbf{x}, t). \quad (10)$$

Using (9) and (10), the following interdependences between the initial and independent velocities and accelerations can be introduced:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{y}} \\ \dot{\mathbf{x}}_D \end{bmatrix} = \begin{bmatrix} I^{(f)} \\ A \end{bmatrix} \dot{\mathbf{y}} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\eta} \end{bmatrix} = D^T \dot{\mathbf{y}} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\eta} \end{bmatrix}, \quad (11)$$

$$\ddot{\mathbf{x}} = D^T \ddot{\mathbf{y}} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\xi} \end{bmatrix}, \quad (12)$$

where $I^{(f)}$ denotes the $f \times f$ identity matrix; and $\mathbf{0}$ is the f -dimensional null vector.

The $f \times e$ matrix $D(\mathbf{x}, t)$ is a priori of maximal rank, and is an orthogonal complement matrix to the constraint matrix C in the e -space of the system's configuration, i.e. $DC^T = \mathbf{0}$. Thus, the columns of D^T are (contravariant) components of vectors $\underline{d}_j (j = 1, \dots, f)$ which span the *tangent* subspace in the e -space. On the other hand, the columns of C^T are (covariant) components of constraint vectors $\underline{c}_i (i = 1, \dots, q)$ which span the *orthogonal* (or *constrained*) subspace. The tangent and orthogonal subspaces complement each other in the e -space, and $DC^T = \mathbf{0}$ expresses the orthogonality conditions $\underline{d}_j \cdot \underline{c}_i = 0 (j = 1(1)f; i = 1(1)q)$. Since of linear independence, $\underline{c}_1, \dots, \underline{c}_q, \underline{d}_1, \dots, \underline{d}_f$ span a new base $e_{cd} = [e_c^T \ e_d^T]^T$ in the e -space, where $e_c = [\underline{c}_1, \dots, \underline{c}_q]^T$ and $e_d = [\underline{d}_1, \dots, \underline{d}_f]^T$ are the base vectors of orthogonal and tangent subspaces, respectively. The transformation formula between the (covariant) bases e_{cd} and e_x is

$$e_{cd} = \begin{bmatrix} e_c \\ e_d \end{bmatrix} = \begin{bmatrix} CM^{-1} \\ D \end{bmatrix} e_x = T_{cd} e_x, \quad (13)$$

where $e_x = [\underline{k}_1, \dots, \underline{k}_e]^T$ are the base vectors spanning the directions of \mathbf{x} . The appearance of M^{-1} in the upper part of T_{cd} comes evident after a little inspection. Since C^T contains covariant components of the base vectors of the orthogonal subspace, the transformation between the covariant base vectors e_c and e_x requires the CM^{-1} formula. On the other hand, D^T contains contravariant components, and the transformation between e_d and e_x is defined by matrix D . For details refer to Blajer [27].

Using the above definitions, the dynamic equations (3) can be projected into the base e_{cd} , which is equivalent to the left-sided premultiplication of these equations by T_{cd} . The tangential projection (into e_d base), after considering (11) and (12), leads

to the minimal set of constraint reaction-free (or canonical) dynamic equations in independent coordinates

$$\mathbf{M}_d(\mathbf{x}, t) \dot{\mathbf{y}} = \mathbf{h}_d(\dot{\mathbf{y}}, \mathbf{x}, t), \quad (14)$$

where

$$\mathbf{M}_d = \mathbf{DMD}^T, \quad (15)$$

$$\mathbf{h}_d = \mathbf{D} \left(\mathbf{h} - \mathbf{M} [\mathbf{0}^T \boldsymbol{\xi}^T]^T \right). \quad (16)$$

As \mathbf{M} is the metric tensor matrix of base \mathbf{e}_r , the metric tensor matrix of base \mathbf{e}_{cd} can be written as

$$\mathbf{M}_{cd} = \mathbf{T}_{cd} \mathbf{M} \mathbf{T}_{cd}^T = \begin{bmatrix} \mathbf{CM}^{-1}\mathbf{C}^T & \mathbf{0} \\ \mathbf{0}^T & \mathbf{DMD}^T \end{bmatrix} = \begin{bmatrix} \mathbf{M}_c & \mathbf{0} \\ \mathbf{0}^T & \mathbf{M}_d \end{bmatrix}, \quad (17)$$

where $\mathbf{M}_c = \mathbf{CM}^{-1}\mathbf{C}^T$ and $\mathbf{M}_d = \mathbf{DMD}^T$ are the metric tensor matrices of bases \mathbf{e}_c and \mathbf{e}_d , respectively; and $\mathbf{0}$ is the $q \times f$ null matrix. The above relation, which will be of use in the following, indicates that the orthogonal and tangent subspaces really complement each other in the e -space.

By appending $\dot{\mathbf{y}} = \mathbf{v}$ to (14), $2f$ first-order differential equations in \mathbf{v} and \mathbf{y} follow. However, since \mathbf{M}_d and \mathbf{h}_d depend on all initial coordinates \mathbf{x} , the constraint equation (4) have to be solved at each step of integration for \mathbf{x}_D as function of the current values of \mathbf{y} , and this process is usually computationally very expensive. To overcome this problem related to the method described, the projection criterion will be presented for a proper choice of the independent coordinates and a symbolical inverse kinematics approach will be proposed.

6. Projective criterion for coordinate partitioning

The projective criterion for coordinate partitioning proposed in this paper deals with a system's configuration space which is not a Cartesian one but an e -dimensional Riemannian space. The norm of a vector in such a space has thus to be redefined according to the vector space algebra. The aspects of contravariant/covariant vector representations are of importance for this definition and for the further base transformations in the e -space, see e.g. Blajer [27]. The transformation matrix \mathbf{T}_{cd} defined in (13) is the mapping of the covariant representations \underline{k}_i^* of vectors $\underline{k}_i = \underline{k}_i^{*T} \mathbf{e}_x^*$ ($i = 1(1)e$),

$$\underline{k}_i^* = [0, \dots, 0, 1, 0, \dots, 0]^T, \quad (18)$$

into e_{cd}^* base, i.e.

$$\underline{k}_i^{*(cd)} = \mathbf{T}_{cd} \underline{k}_i^* = \begin{bmatrix} CM^{-1} \\ D \end{bmatrix} \underline{k}_i^*. \quad (19)$$

The vector \underline{k}_i defined this way can be interpreted as a unit vector along \dot{x}_i direction, $\underline{\dot{x}} \cdot \underline{k}_i = \dot{x}^T \underline{k}_i^* = \dot{x}_i$, and this elucidate its (covariant) representation in (18). Then, it comes from (19) that the i -th column of CM^{-1} is the (covariant) representation of \underline{k}_i in e_c^* base, whereas the i -th column of D is the (covariant) representation of \underline{k}_i in e_d^* base. Denoting these representations by $\underline{k}_i^{*(c)}$ and $\underline{k}_i^{*(d)}$, respectively, it can be written that:

$$\begin{aligned} CM^{-1} &= [\underline{k}_1^{*(c)} \ \underline{k}_2^{*(c)} \dots \underline{k}_e^{*(c)}]_{(q \times e)}, \\ D &= [\underline{k}_1^{*(d)} \ \underline{k}_2^{*(d)} \dots \underline{k}_e^{*(d)}]_{(f \times e)}, \end{aligned} \quad (20)$$

i.e. $\underline{k}_i^{*(c)}$ and $\underline{k}_i^{*(d)}$ are the i -th columns of CM^{-1} and D , respectively.

Using generalized scalar products, $|\underline{k}_i|^2$, $|\underline{k}_i^{*(c)}|^2$ and $|\underline{k}_i^{*(d)}|^2$ can be written as follows:

$$\begin{aligned} |\underline{k}_i|^2 &= \underline{k}_i^{*T} M^{-1} \underline{k}_i^* = M^{-1}(i, i), \\ |\underline{k}_i^{*(c)}|^2 &= (\underline{k}_i^{*(c)})^T M_c^{-1} \underline{k}_i^{*(c)}, \\ |\underline{k}_i^{*(d)}|^2 &= (\underline{k}_i^{*(d)})^T M_d^{-1} \underline{k}_i^{*(d)}, \end{aligned} \quad (21)$$

where $M^{-1}(i, i)$ is the i th entry of M^{-1} ; and M_c and M_d are defined in (15). Basing on (21), the following generalized formulation of the projective criterion for coordinate partitioning can be introduced:

$$\cos^2 \alpha_i = \frac{|\underline{k}_i^{*(d)}|^2}{|\underline{k}_i|^2} = \frac{(\underline{k}_i^{*(d)})^T M_d^{-1} \underline{k}_i^{*(d)}}{M^{-1}(i, i)}, \quad (22a)$$

$$\cos^2 \beta_i = \frac{|\underline{k}_i^{*(c)}|^2}{|\underline{k}_i|^2} = \frac{(\underline{k}_i^{*(c)})^T M_c^{-1} \underline{k}_i^{*(c)}}{M^{-1}(i, i)}. \quad (22b)$$

The bigger $\cos^2 \alpha_i$ (the smaller $\cos^2 \beta_i$) the closer is \underline{k}_i to the tangent hyperplane and the better x_i as an independent coordinate.

In fact two formulae for the reported criterion have been introduced, (22a) and (22b). Respectively, they express the squared cosines (generalized to the e -spaces) of angels

between the vector \underline{k}_i and its projections $\underline{k}_i^{(d)}$ and $\underline{k}_i^{(c)}$ into the tangent and orthogonal subspaces. The matrix M_d used in (22a) is actually the mass matrix of the minimal-dimension dynamic equations (14), and thus is available (more or less explicitly) in its inverted form at each instant of the system motion simulation. The matrix $M_c = CM^{-1}C^T$ used in (22b) has to be formulated and inverted individually. Therefore, the formulation (22a) is recommendable for the reported formulation.

For the current set y , the reported criterion can be applied occasionally to check or redefine the choice for y as related those components of x whose corresponding $\cos^2 \alpha_i$ ($i = 1, \dots, e$) have the biggest values.

7. Application of inverse kinematics algorithms

The essential shortcoming of the coordinate partitioning method is the necessity of inverting C_D in order to determine $A = -C_D^{-1}C_I$, $\eta = C_D^{-1}a$, and $\xi = C_D^{-1}b$, required for the formulation of equations (12) or (17). During the simulation process C_D has to be inverted at each step of integration, and this may bring some inefficiency in computations.

In this section advantages are emphasized that may arise in the coordinate partitioning approach to the dynamic analysis of constrained mechanical systems by adapting special algorithms of inverse kinematics developed in the field of robotics, and of remarkable importance is a technique developed by Woernle [30]. According to this technique, the kinematic chains are parted into two open chains so that to select relations with a reduced number of unknowns. Then, setting some coordinates to be *frozen* (independent), the recursive relations for the other (dependent) coordinates as function of the *frozen* ones are found without introducing the constraint equations in the form (4), see also Eppinger and Kreuzer [31], and Blajer, Schiehlen and Schirm [32], and Schiehlen and Blajer [33]. These recursive relations are denoted symbolically as

$$\underline{x}_D = \underline{x}_D(y, t), \quad (23)$$

and are recognized also as *closing conditions*, Ref. [34]. In fact, (23) are often quite complex, and the amount of labour required for their derivation depends greatly on the skill of the investigator in using the inverse kinematics procedures. Nevertheless, this initial work pays in the further analysis. The (recursive) relations for (9) and (10) are usually not so laborious to be obtained analytically. They can also be derived by using computer symbolical formalisms like NEWEUL [8], [21].

The application of inverse kinematics algorithms benefits in analytical (though recursive) formulae for \mathbf{x}_D , \mathbf{A} , $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$. This accelerates usually the numerical formulation of the tangent dynamic equations (14), and the final governing equations of motion can be written in the following simplified 2f-order form:

$$\dot{\mathbf{M}}_d(\mathbf{y}) \dot{\mathbf{v}} = \hat{\mathbf{h}}_d(\mathbf{v}, \mathbf{y}, t), \quad (24a)$$

$$\dot{\mathbf{y}} = \mathbf{v} \quad (24b)$$

where $\dot{\mathbf{M}}_d$ and $\hat{\mathbf{h}}_d$ correspond to \mathbf{M}_d and \mathbf{h}_d defined in (15) and (16) after substituting $\mathbf{x} = [\mathbf{y}^T \quad \mathbf{x}_D^T(\mathbf{y}, t)]^T$ and $\dot{\mathbf{y}} = [\mathbf{v}^T \quad (\mathbf{A}(\mathbf{y}, t)\mathbf{v} + \boldsymbol{\eta}(\mathbf{y}, t))^T]^T$, where $\mathbf{x}_D(\mathbf{y}, t)$, $\mathbf{A}(\mathbf{y}, t)$, $\boldsymbol{\eta}(\mathbf{y}, t)$ and $\boldsymbol{\xi}(\mathbf{v}, \mathbf{y}, t)$ represent the recursive formulae from the inverse kinematics. Note that the closing conditions (23) replace the constraint equations (4), i.e. it can be written

$$\hat{\Phi}(\mathbf{x}, t) = -\mathbf{x}_D(\mathbf{y}, t) + \mathbf{x}_D = 0. \quad (25)$$

Thus, the solution of (24) is released from the problem of constraint violation. Note also that, as all the entries of $\dot{\mathbf{x}}$ and \mathbf{x} are determined at each step of integration of (24), an eventual transition from one set of $\dot{\mathbf{y}}$ to another will not yield any inconsistency in the initial value problem of accordingly reformulated governing equations. Obviously, an appropriate number of recursive formulae (23), (9) and (10) for different possible (or all) sets of \mathbf{y} from \mathbf{x} has to be prepared in advance.

Consider the planar four-bar linkage shown in Fig.9. In order to build an open-loop system, each of the joints O_i ($i = 1(1)4$) can be cut. The coordinates \mathbf{x} of

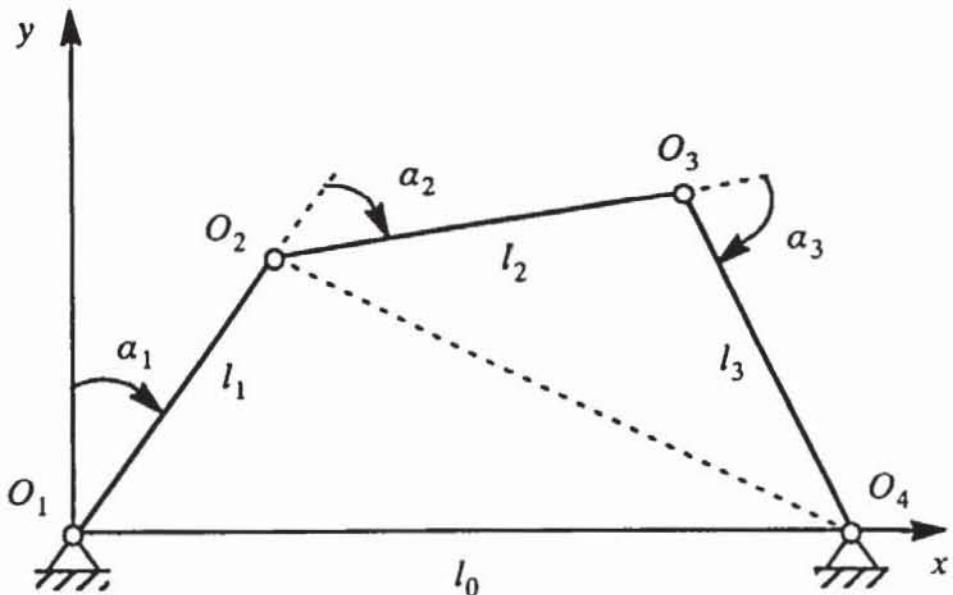


Figure 9: Four-bar mechanism

the *unconstrained* system can also be defined differently. In the example case, the mechanism was cut in joint O_4 and the relative coordinates $\mathbf{z} = [\alpha_1, \alpha_2, \alpha_3]^T$ have been chosen to represent a planar manipulator with the end-effector fixed in point O_4 . The dynamic equations of the system, corresponding to (14), will not be reported here.

The constraints of the system can be expressed either implicitly by constraint equations (4) or explicitly by closing conditions (23), which yields respective formulations of matrix D defining the tangent subspace. In the following an application of inverse kinematics algorithms leading to recursive relations for the closing conditions will be demonstrated. The subsequent derivation for the simple example bases on the approach given by Woernle [30]. According to the approach, the mechanism is separated, by cutting in joints O_2 and O_4 , into the *lower* and *upper* segments, and the closure condition is

$$\vec{\mathbf{r}}_{lower} \circ \vec{\mathbf{r}}_{lower} - \vec{\mathbf{r}}_{upper} \circ \vec{\mathbf{r}}_{upper} = 0 , \quad (26)$$

where $\vec{\mathbf{r}}_{lower} := O_1O_4 - O_1O_2$, $\vec{\mathbf{r}}_{upper} := O_3O_2 - O_3O_4$. Due to the used segmentation, (26) depends on α_1 and α_3 (does not depend on α_2), and only one of these coordinates can be chosen for an independent one in the subsequent derivation (the choice $\mathbf{y} = [\alpha_2]$ would require a different segmentation). Here, the relations (23), (11) and (12) are reported only for $\mathbf{y} = [\alpha_1]$.

Solving (26), one obtaines

$$\alpha_3 = \pm \arccos \left(\frac{l_0^2 + l_1^2 - l_2^2 - l_3^2 - 2l_0l_1 \sin \alpha_1}{2l_2l_3} \right) . \quad (27)$$

The complementary relation for α_2 is obtained then as suggested in [30] from

$$\begin{aligned} \sin \alpha_2 &= \frac{l_0l_2 \cos \alpha_1 + l_0l_3 \cos(\alpha_1 + \alpha_3) + l_1l_3 \sin \alpha_3}{l_0^2 + l_1^2 - 2l_0l_1 \sin \alpha_1} , \\ \cos \alpha_2 &= \frac{l_0l_2 \sin \alpha_1 + l_0l_3 \sin(\alpha_1 + \alpha_3) - l_1l_3 \cos \alpha_3 - l_1l_2}{l_0^2 + l_1^2 - 2l_0l_1 \sin \alpha_1} . \end{aligned} \quad (28)$$

The relations (27) and (28) express recursively $\alpha_3(\alpha_1)$ and $\alpha_2(\alpha_1)$ as defined in (23). Differentiation of these closing conditions leads to:

$$\begin{aligned} \dot{\alpha}_3 &= \frac{l_0l_1 \cos \alpha_1}{l_2l_3 \sin \alpha_3} \dot{\alpha}_1 , \\ \dot{\alpha}_2 &= -\frac{l_0 \cos(\alpha_1 + \alpha_2)}{l_3 \sin \alpha_3} \dot{\alpha}_1 - \dot{\alpha}_3 , \end{aligned} \quad (29)$$

and therefore, the matrix D defined in (11) can be stated as

$$D = \begin{bmatrix} 1 & -\frac{l_0 l_1 \cos \alpha_1 + l_0 l_2 \cos(\alpha_1 + \alpha_2)}{l_2 l_3 \sin \alpha_3} & \frac{l_0 l_1}{l_2 l_3} \cdot \frac{\cos \alpha_1}{\sin \alpha_3} \end{bmatrix}. \quad (30)$$

After differentiating (29) the vector ξ introduced in (12) can be formed as shown by Blajer, Schiehlen and Schirm [32].

Now the problem of the *best* independent coordinate choice is discussed. For particular linkage data, the results obtained by using the projective criterion are shown in Fig.10. The linkage geometry was set to assure that the choice $y = [\alpha_1]$ never leads to a singularity, enabling one to plot the results throughout the whole range $\alpha_1 \in [0, 2\pi]$. As seen, both α_1 and α_2 are acceptable choices for y in any linkage configuration, none of them, however, can be assigned to be the *best* independent coordinate over the whole range of α_1 . It is evident, also, that the choice $y = [\alpha_3]$ is the worst as leading to singularities at $\alpha_1 = \pi/2$ and $\alpha_1 = 3\pi/2$, and due to relatively small values of tangent projections.

The numerical simulation can now be carried out in the *best* independent coordinate according to the projective criterion, i.e. changes between the coordinates α_1 and α_2 are necessary to ensure the integration with the *best* coordinate. This change can be done without a loss in integration order and stepsize by using a modified Adams-Bashforth integration code.

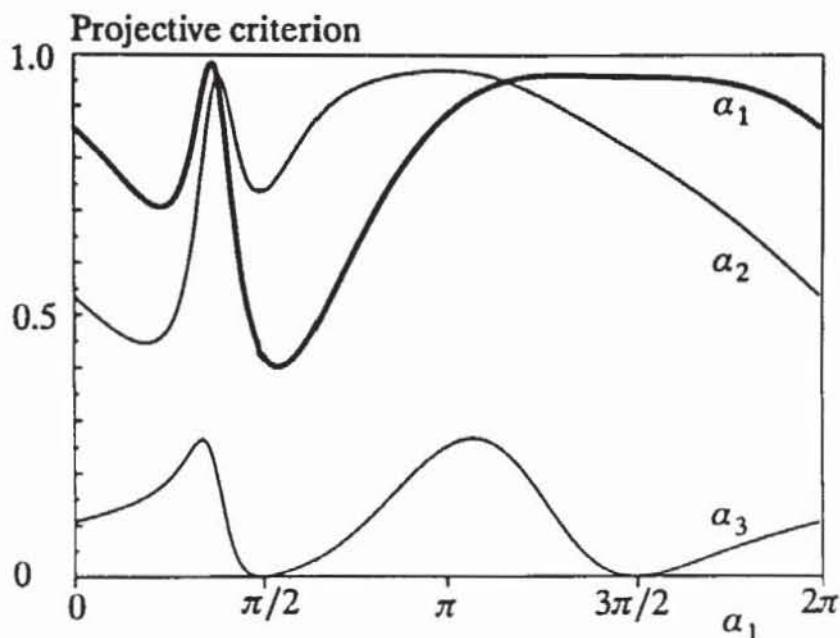


Figure 10: Projective criterion (four-bar mechanism)

8. Visualization of simulation results

A convenient verification a dynamic visualization of a multibody system simulation is obtained by a 3D computer graphics animation. Animation methods differ according to the geometry model, rendering algorithms and possible user interaction. The most sophisticated animation method is achieved by rendering algorithms like raytracing and radiosity. These rendering techniques result in realistic images, but suffer from time-consuming computations. During image display, no interactive modification of the view projection is possible. A raytraced image of the crank-slider mechanism is shown in Figure 11.

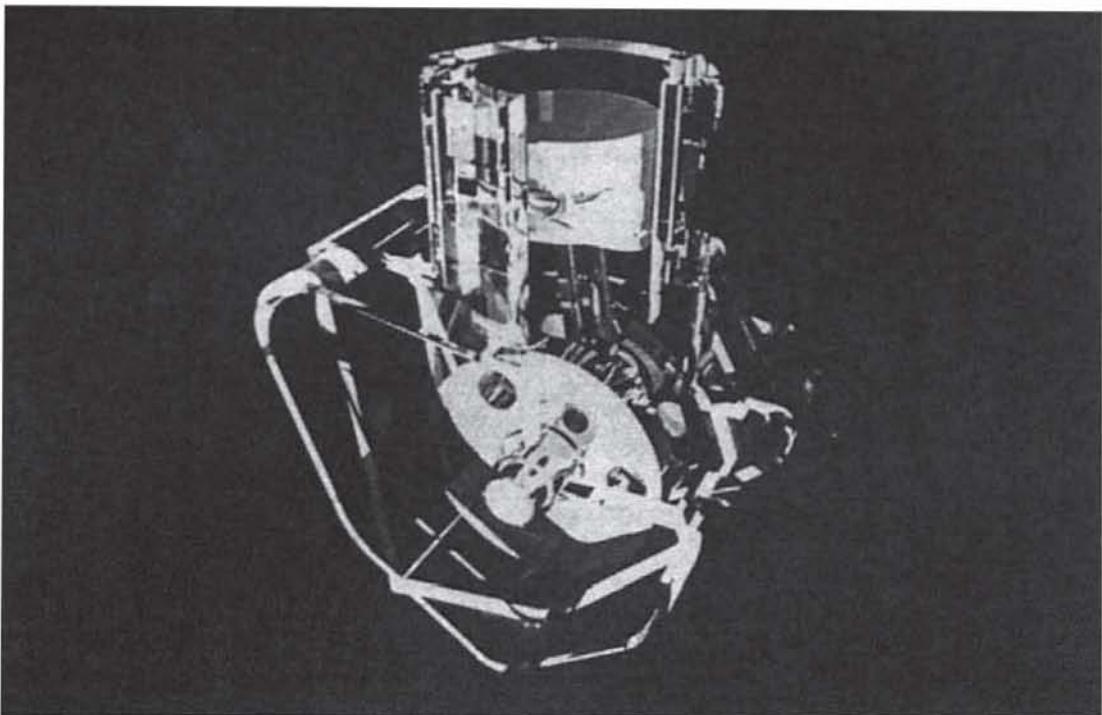


Figure 11: Raytracing of crank slider mechanism

Most CAD-3D-systems offer modules for the generation of images with hidden line and hidden surfaces removal and shaded surfaces. Often, the solid model and rendering algorithms yield sophisticated 2D drawings for documentation purposes, but allow a dynamic visualization only in a wireframe mode.

Consequently the unified approach to display a broad variety of simulation result for different initial conditions, visualization systems and applications is based on the planar face model. The visualization module VISANI for the interactive, high speed animation of arbitrary multibody systems is described by Daberkow [17]. As a result of the simulation, a time plot of the crankshaft bearing force of the mechanism under

an applied piston gas force and an animated sequence is shown in Figure 12.

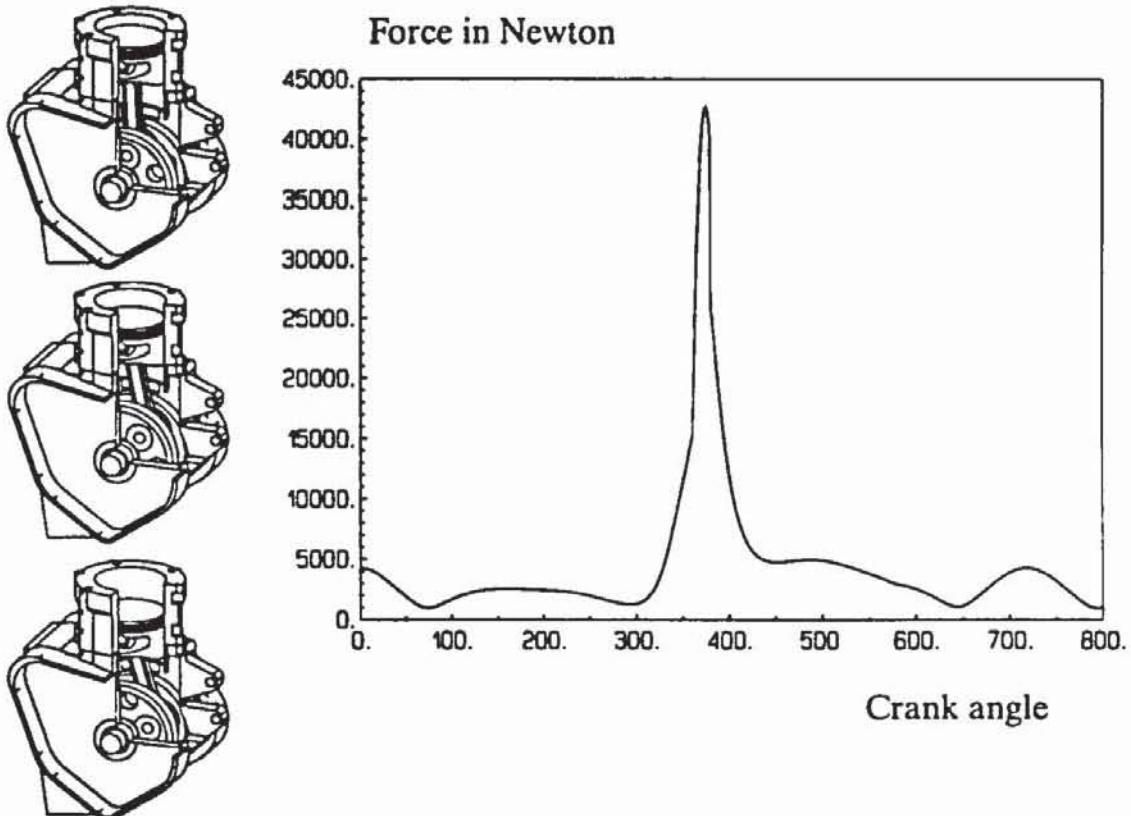


Figure 12: Time plot and animated sequences of the crank slider mechanism

9. Flexible multibody systems

A free rigid body i has $e_{is} = 6$ degrees of freedom which is also true for the reference frame of an elastic body in its undeformed configuration. The deformation of a flexible body depends on a displacement field $\mathbf{u} = \mathbf{u}(\mathbf{c}, t)$ characterizing the position of a material particle identified by the vector \mathbf{c} in the undeformed configuration. Assuming a discretization of the elastic body, the displacement field is expressed as a linear combination of selected deformation modes

$$\mathbf{u}(\mathbf{c}, t) = \Phi(\mathbf{c})\mathbf{q}(t) \quad (31)$$

where the vector $\mathbf{q}(t) = [q_1, \dots, q_{e,f}]^T$ represents the e_{if} flexible coordinates of the body i and $\Phi(\mathbf{c})$ means the corresponding shape function vector of the deformation modes. The flexible coordinates enlarge the number of degrees of freedom of body i to $e_i = e_{is} + e_{if}$. The same is true for the total multibody system with p bodies, $i = 1(1)p$. On the other hand, the g holonomic constraints reduce the number of

degrees of freedom. As result, one obtains for the flexible system's degrees of freedom $f = e_s - q + e_f$. Then, the global $f \times 1$ position vector of generalized coordinates can be partitioned as

$$\mathbf{y}_q = [\mathbf{y}^T \ \mathbf{q}^T]^T \quad (32)$$

where the $(e_s - q) \times 1$ vector \mathbf{y} denotes the rigid body motion and the $e_f \times 1$ vector \mathbf{q} describes the elastic coordinates of the system. Further, it is assumed that the elastic coordinates are small compared to the rigid body motion of the system.

The equations of motion of the flexible multibody systems are an extension of (24) and result in

$$\begin{bmatrix} \dot{\mathbf{M}}_d(\mathbf{y}) & \dot{\mathbf{M}}_{de}(\mathbf{y}_q) \\ \dot{\mathbf{M}}_{de}^T(\mathbf{y}_q) & \dot{\mathbf{M}}_e \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{y}} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \dot{\mathbf{K}}_e \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{q} \end{bmatrix} = \hat{\mathbf{h}}_q(\mathbf{y}_q, \dot{\mathbf{y}}_q, t) \quad (33)$$

where the $e_s \times e_f$ matrix $\dot{\mathbf{M}}_{de}$ represents the coupling between flexible and rigid body motion and the $e_f \times e_f$ matrices $\dot{\mathbf{M}}_e$ and $\dot{\mathbf{K}}_e$ characterizes the small structural vibrations of the system.

For a more detailed evaluation it requires a proper choice of the deformation modes characterized by the shape functions $\Phi(\mathbf{c})$, see (31). An approach closely related to rigid body modeling uses the rigid-elastic superelement, Fig. 13, and was presented by Rauh [35].

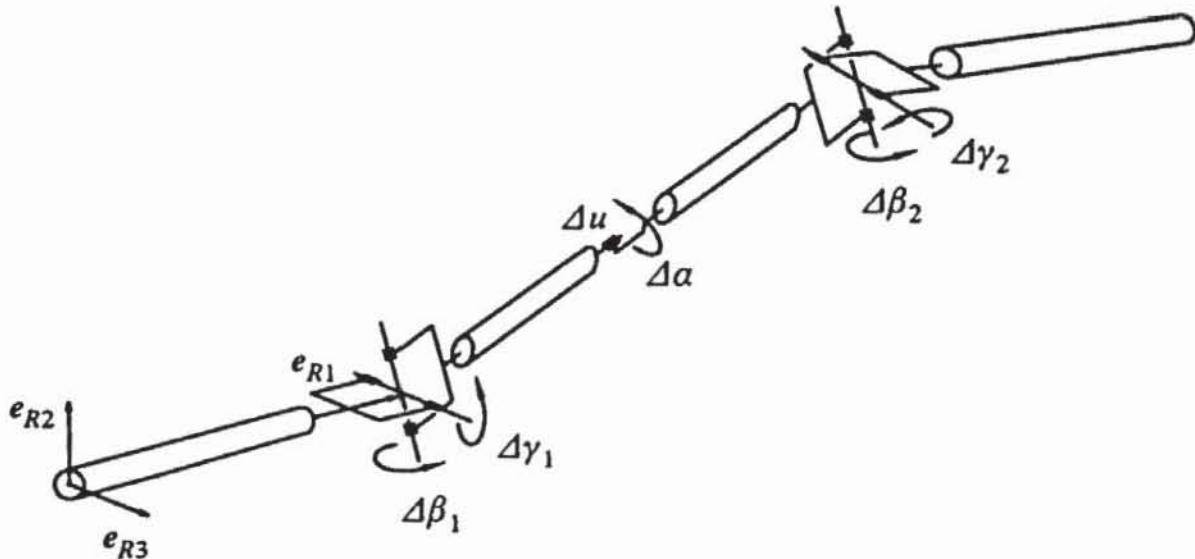


Figure 13: Rigid-elastic superelement for a flexible beam

The rigid-elastic superelement represents a piecewise constant shape function and the elastic properties result in discrete spring coefficients with respect to the elastic

coordinates $\Delta\alpha$, $\Delta\beta_1$, $\Delta\beta_2$, $\Delta\gamma_1$, $\Delta\gamma_2$ and Δu . The advantage of this approach is that standard rigid body codes like NEWEUL can be applied for the generation of the equations of motion.

Another approach related to the finite element method uses standard finite elements, Fig. 14, and was presented by Bremer and Pfeiffer [36], Melzer [37] and Sorge [38]. It turns out that most of the volume integrals summarized in (33) are time-invariant

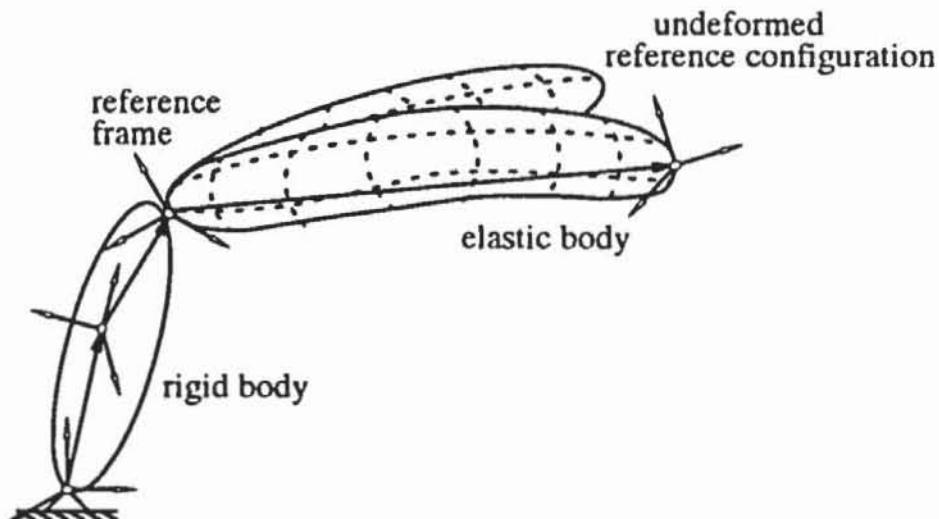


Figure 14: Finite element representation of an elastic body

and may be precomputed, see Melzer [37]. This means that flexible multibody systems with small elastic displacements can be analysed using symbolic computations. Compared to purely numerical computations by the standard finite element method, the symbolic computations are much more efficient.

10. Optimization of multibody systems

Optimization is considered as an integrated modeling and parameter evaluation process for multibody systems. Computer aided generation of symbolic equations of motion and sensitivity analysis helps to cut down the time for designing such systems. The method will be presented for vehicle suspension systems following Bestle, Eberhard and Schiehlen [39]

Designing a mechanical system by computer aided engineering involves several steps: (i) modeling, (ii) choosing design variables, (iii) defining performance criteria and constraints, (iv) solving the optimization problem resulting in optimal values for the design variables. Finally, the optimal system has to be realized, and there may occur some economic constraints requiring a reformulation of the design problem. This may

cause changes in the model, in the choice of the design variables, and in definitions of performance criteria, or it requires additional constraints.

Multibody system models are well accepted for simulating the dynamic behavior of vehicle systems in a frequency domain less than 50Hz. For low frequency motions the carbody, the driver, and the axles can be considered as rigid bodies connected by ideal links and coupled by force elements like springs, dampers, and active elements, Fig. 15.

Springs in a suspension system act on both the carbody and the axles. For linear springs the force proportional to the relative displacement reads as

$$F_c = c_F(z_P - z_Q) \quad (34)$$

where c_F is the stiffness coefficient. The force of a linear damper is proportional to the relative velocity:

$$F_d = d_F(\dot{z}_P - \dot{z}_Q) . \quad (35)$$

For improved modeling nonlinear characteristics like a cubic damper may be used, too:

$$F_d = d_F(\dot{z}_P - \dot{z}_Q)^3 . \quad (36)$$

For more effective damping of the carbody motions forces proportional to the absolute velocity of the carbody, i.e.

$$F_d = d_A \dot{z}_P , \quad (37)$$

may be better suited than a relative damper (35). Such a device is called a *skyhook damper* which has to be realized by an active element.

To some extent, the designer of a suspension system has a free choice of the values of the stiffness coefficients c_F , damping coefficients d_F , and control parameters d_A . Any of these parameters may be chosen as a design variable. But geometrical data like the height of the center of gravity of the carbody or the masses and moments of inertia of some of the bodies may be changed within given ranges, too. This will result in a different behavior due to changes in the inertia matrix and the generalized forces, respectively.

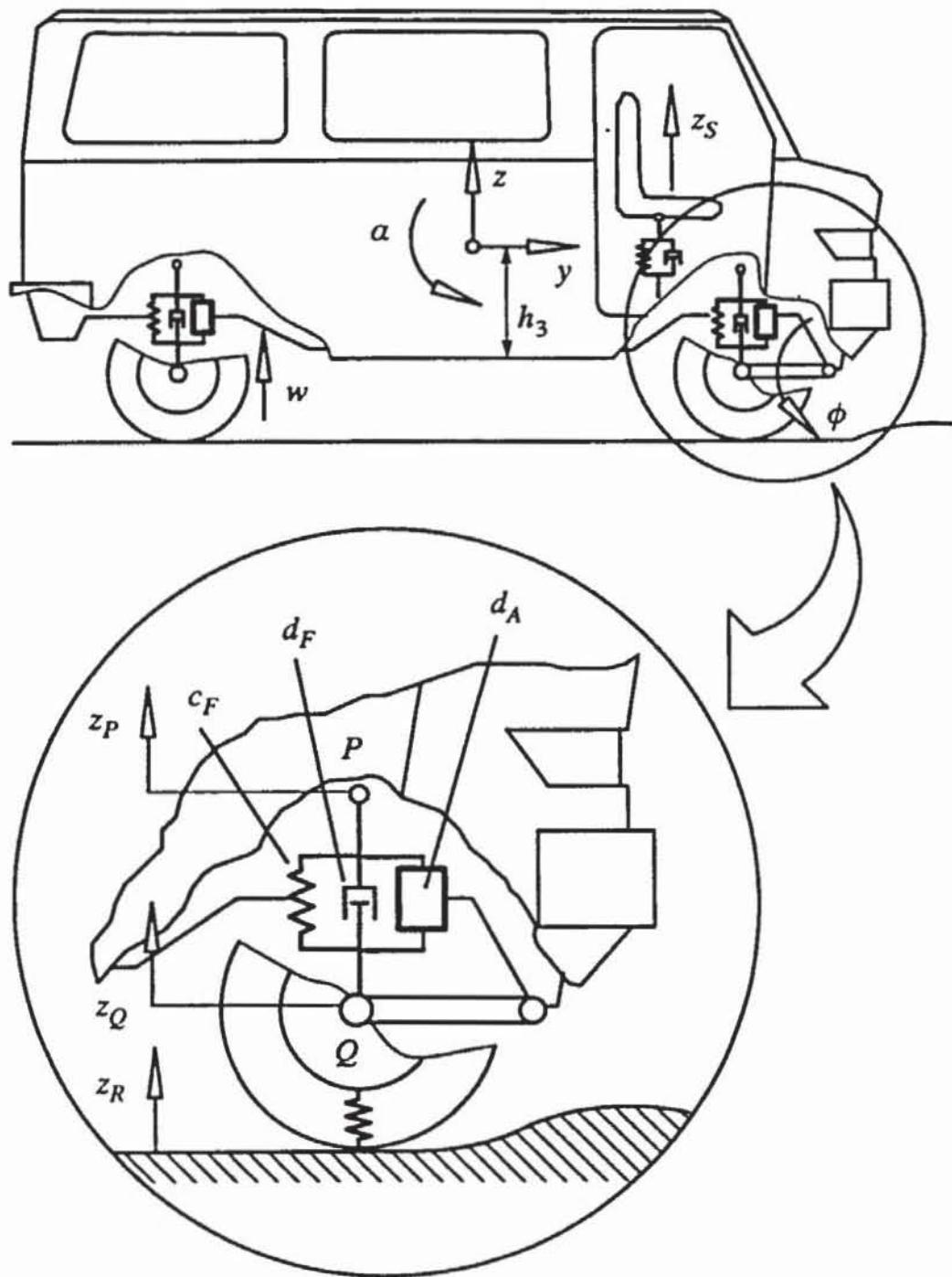


Figure 15: Vehicle modeled for plane motion

After summarizing all design variables in a $h \times 1$ vector \mathbf{p} , the equations of motion (24) have to be rewritten as

$$\mathbf{M}(\mathbf{y}, \mathbf{p}) \dot{\mathbf{v}} = \mathbf{h}(\mathbf{v}, \mathbf{y}, t, \mathbf{p}), \quad (38a)$$

$$\dot{\mathbf{y}} = \mathbf{v} \quad (38b)$$

These equations are supplemented by initial conditions for the state coordinates \mathbf{y} and \mathbf{v} at some given time t^0 :

$$\begin{aligned} \mathbf{y}^0 := \mathbf{y}(t^0) : \Phi^0(t^0, \mathbf{y}^0, \mathbf{p}) &= 0, & \det \frac{\partial \Phi^0}{\partial \mathbf{y}^0} &\neq 0, \\ \mathbf{v}^0 := \mathbf{v}(t^0) : \dot{\Phi}^0(t^0, \mathbf{y}^0, \mathbf{v}^0, \mathbf{p}) &= 0, & \det \frac{\partial \dot{\Phi}^0}{\partial \mathbf{v}^0} &\neq 0, \end{aligned} \quad (39)$$

where the Jacobians of Φ^0 and $\dot{\Phi}^0$ have to be regular for determining the initial state \mathbf{y}^0 and \mathbf{v}^0 uniquely.

The application of systematic methods for searching optimal values of the design variables requires the definition of performance criteria. An important function of suspension systems is to improve the riding comfort by isolating the carbody from roadway unevenness. The comfort of a vehicle can be evaluated by the acceleration \ddot{z}_s , acting on the driver. If the driving over a bump is considered as a test, accelerations may be penalized by the square of the time, too. An integral performance criterion can then be expressed by

$$\psi_C = \int_{t^0}^{t^1} t^2 \ddot{z}_s^2 dt. \quad (40)$$

Another important task of a suspension system is to provide safety which is related to the dynamic variation in the load between the wheels and the road. If the tire is considered as a linear spring, the load is proportional to the relative displacement between wheel and road surface, Fig. 15. A performance function evaluating the safety of a vehicle is

$$\psi_S = \int_{t^0}^{t^1} (z_Q - z_R)^2 dt. \quad (41)$$

Carbody isolation results in very soft suspensions, but this will also yield large relative motions between carbody and axles. For limiting the relative displacements, a criterion like

$$\psi_D = \int_{t^0}^{t^1} \left(\frac{z_P - z_Q}{s_0} \right)^6 dt. \quad (42)$$

may be used where s_0 is a predefined amplitude following from the vehicle design, which should not be exceeded too much.

The final time t^1 in the criteria (40)-(42) may be defined by the user or given implicitly by some final state $\mathbf{y}^1 = \mathbf{y}^1(t^1)$, $\mathbf{v}^1 = \mathbf{v}^1(t^1)$:

$$\begin{aligned} t^1 &: H^1(t^1, \mathbf{y}^1, \mathbf{v}^1, \mathbf{p}) = 0, \\ \dot{H}^1 &:= \frac{dH^1}{dt^1} = \frac{\partial H^1}{\partial t^1} + \left(\frac{\partial H^1}{\partial \mathbf{y}^1} \right)^T \mathbf{v}^1 + \left(\frac{\partial H^1}{\partial \mathbf{v}^1} \right)^T \dot{\mathbf{v}}^1 \neq 0. \end{aligned} \quad (43)$$

E.g., for the vehicle excited by a bump the final time has to be chosen sufficiently large to guarantee substantial decrease of vibrations.

The problem as stated can be reduced to a nonlinear programming problem. If some value is assigned to each design variable, the state $\mathbf{y}(t)$, $\mathbf{v}(t)$ is completely determined by the equations of motion (38) and initial conditions (39). Further, the final time is given by equation (43). Then, the criteria (40)-(42) are functions of the design variables only. In engineering problems, there are always some restrictions on the design variables. If there are only lower and upper bounds, the feasible design space is given by

$$P = (\mathbf{p} \in R^h | p_i^l \leq p_i \leq p_i^u, i = 1(1)h) . \quad (44)$$

Minimizing each of the criteria (40)-(42) individually represents a nonlinear programming problem which may be solved by any general purpose optimization algorithm. In general, the optimal points in the design space, i.e.

$$\begin{aligned} \mathbf{p}_C &: \psi_C^* := \psi_C(\mathbf{p}_C) = \min_{\mathbf{p} \in P} \psi_C(\mathbf{p}), \\ \mathbf{p}_S &: \psi_S^* := \psi_S(\mathbf{p}_S) = \min_{\mathbf{p} \in P} \psi_S(\mathbf{p}), \\ \mathbf{p}_D &: \psi_D^* := \psi_D(\mathbf{p}_D) = \min_{\mathbf{p} \in P} \psi_D(\mathbf{p}), \end{aligned} \quad (45)$$

will be different from each other. A feasible design where all three criteria have simultaneously minimal values does not exist in general. E.g., high riding comfort requires a very soft suspension whereas low relative displacement between carbody and axles can only be achieved by a stiff suspension. Multicriteria optimization theory, see e.g. Osyczka [40] offers optimal designs in such conflicting situations, too.

An often used method is the weighted objectives method. Instead of the individual criteria a scalar weighted-sum criterion has to be minimized:

$$\psi = w_C \frac{\psi_C}{\bar{\psi}_C} + w_S \frac{\psi_S}{\bar{\psi}_S} + w_D \frac{\psi_D}{\bar{\psi}_D}, \quad w_C + w_S + w_D = 1. \quad (46)$$

The numerical procedure for solving such problems is an iterative process starting from an user-defined design and finding a better one step by step. The application of such procedures for the optimization of vehicle systems is very time consuming. In each step the criterion (46) has to be evaluated which requires the numerical integration of the equations of motion. Therefore, rapidly converging optimization algorithms like the sequential quadratic programming (SQP) method is recommended, see Fletcher [41].

Advanced optimization algorithms like SQP are based not only on function evaluations but also on gradient information of the objective and constraint functions. Since the gradients cannot be computed analytically for engineering systems, some algorithms use the possibility of computing gradients by numerical differentiation, e.g. with the forward difference formula:

$$\nabla \psi_k \approx \frac{\psi(p + \Delta p_k e_k) - \psi(p)}{\Delta p_k}, \quad k = 1(1)h, \quad (47)$$

where $\Delta p_k e_k$ is a small perturbation of the design variable p_k . But applying difference formulas to the optimization of multibody systems will yield poor results due to the rather large error in evaluating ψ by numerical integration. Therefore, it is advisable to use sensitivity analysis methods for generating analytical information on the gradients.

For the sensitivity analysis of integral type functions like equations (40)-(42) two methods have been developed as reported by Haug [42]: the direct differentiation method and the adjoint variable method. From a computational point of view the latter is preferable and will be discussed here. For a general criterion

$$\psi(p) = G^1(t^1, y^1, v^1, p) + \int_{t_0}^{t^1} F(t, y, v, \dot{v}, p) dt \quad (48)$$

the gradient can be computed as

$$\begin{aligned} \nabla \psi = \frac{d\psi}{dp} &= \frac{\partial G^1}{\partial p} - \tau^1 \frac{\partial H^1}{\partial p} - \left(\frac{\partial \Phi^0}{\partial p} \right)^T \zeta^0 - \left(\frac{\partial \dot{\Phi}^0}{\partial p} \right)^T \eta^0 \\ &+ \int_{t_0}^{t^1} \left[\frac{\partial F}{\partial p} - \left(\frac{\partial(M\dot{v} - h)}{\partial p} \right)^T (\nu - \xi) \right] dt \end{aligned} \quad (49)$$

where the adjoint variables τ^1 , $\mu(t)$, $\nu(t)$, $\xi(t)$, ζ^0 , η^0 have to be obtained from algebraic and differential equations, see Bestle and Eberhard [43]. The variable τ^1 and the final values of $\mu(t^1)$ and $\nu(t^1)$ are given by

$$\begin{aligned}\tau^1 &= \frac{\dot{G}^1 + F^1}{\dot{H}^1}, \\ \mu(t^1) &= \frac{\partial G^1}{\partial y^1} - \tau^1 \frac{\partial H^1}{\partial y^1}, \\ M^1 \nu(t^1) &= \frac{\partial G^1}{\partial v^1} - \tau^1 \frac{\partial H^1}{\partial v^1}.\end{aligned}\tag{50}$$

The functions $\mu(t)$ and $\nu(t)$ have to be computed by numerical backward integration of the adjoint differential equations

$$\begin{aligned}\dot{\mu} &= \left(\frac{\partial(M\dot{v} - h)}{\partial y} \right)^T (\nu + \xi) - \frac{\partial F}{\partial y}, \\ M\dot{\nu} &= -\mu - \dot{M}\nu - \left(\frac{\partial h}{\partial v} \right)^T (\nu + \xi) - \frac{\partial F}{\partial v}\end{aligned}\tag{51}$$

where the auxiliary variables $\xi(t)$ have to be obtained simultaneously from the linear algebraic equations

$$M\xi = \frac{\partial F}{\partial v}\tag{52}$$

Finally, η^0 and ζ^0 are available successively from the linear algebraic equations

$$\left(\frac{\partial \dot{\Phi}^0}{\partial v^0} \right)^T \eta^0 = M^0 \nu(t^0), \quad \left(\frac{\partial \Phi^0}{\partial y^0} \right)^T \zeta^0 = \mu(t^0) - \left(\frac{\partial \dot{\Phi}^0}{\partial y^0} \right)^T \eta^0.\tag{53}$$

Further studies have shown the adjoint variable method to be suitable for symbolic computations, too. In particular, the partial derivatives with respect to the design variables and the state variables are computed automatically by the programmable formula manipulation package MAPLE [44]. The resulting equations of motion and the adjoint differential equations are then solved by numerical integration. Because of the complexity of these equations and the dependence of the adjoint equations on state variables it is advantageous to use multistep integration algorithms and a corresponding interpolation scheme, as shown by Bestle and Eberhard [43].

The planar model of a vehicle, Fig. 15, consists of 4 rigid bodies. The 6×1 vector \mathbf{y} summarizing the generalized coordinates reads as

$$\mathbf{y} = [y, z, \alpha, \phi, w, z_S]^T. \quad (54)$$

The dynamic behavior will be described by a set of twelve first order differential equations of motion. The vehicle is assumed to drive with a constant velocity of $20m/s$ over a sinusoidal bump of height $0.1m$ and length $3m$, often found in residential areas as "sleeping policemen". The vehicle suspension is modeled by force elements (34)-(37) in parallel configuration. If the control parameter d_A is zero, the suspension is called passive, otherwise it is active. The dampers may be linear (34) or have a progressive characteristic (35).

In the following, the damping and stiffness coefficients of the front and rear suspension, i.e. d_F, d_R, c_F, c_R , the height of the center of gravity h_3 , and in case of active suspensions the control parameter d_A are chosen as design variables. They can be summarized in the 6×1 vector of design variables

$$\mathbf{p} = [d_F, d_R, c_F, c_R, h_3, d_A]^T. \quad (55)$$

Optimization of the vehicle has been performed using criterion (46) with different sets of weighting factors w_c, w_s, w_d . Fig. 16 shows the vertical acceleration of the driver for the initial and the optimized design with respect to criterion (40) using the planar car model with progressive dampers.

For both active and passive suspensions optimization leads to an improved riding comfort. In particular, the active suspension reduces the maximum acceleration as well as long term vibrations.

Fig. 17 shows the riding comfort of an optimized design using criteria (40)-(42) one after the other and linear dampers. Considering the relative displacement leads to high maximal accelerations while optimization of the riding safety yields low but poorly damped accelerations. The analysis of conflicting optimization criteria shows that the improvement of one criterion worsens the other criteria. Thus, only a multi-criteria approach will give an engineering trade-off, Fig. 18. E.g., if a weighted-sum criterion is used instead of the comfort criterion (40) only, the improvement in riding comfort is not as high as in Fig. 18. However, the final result shows a strong improvement of the riding comfort compared to the initial situation without much pay-off in riding safety or relative displacement. It will be a matter of engineering intuition to make a good choice on the weighting coefficients w_c, w_s, w_d .

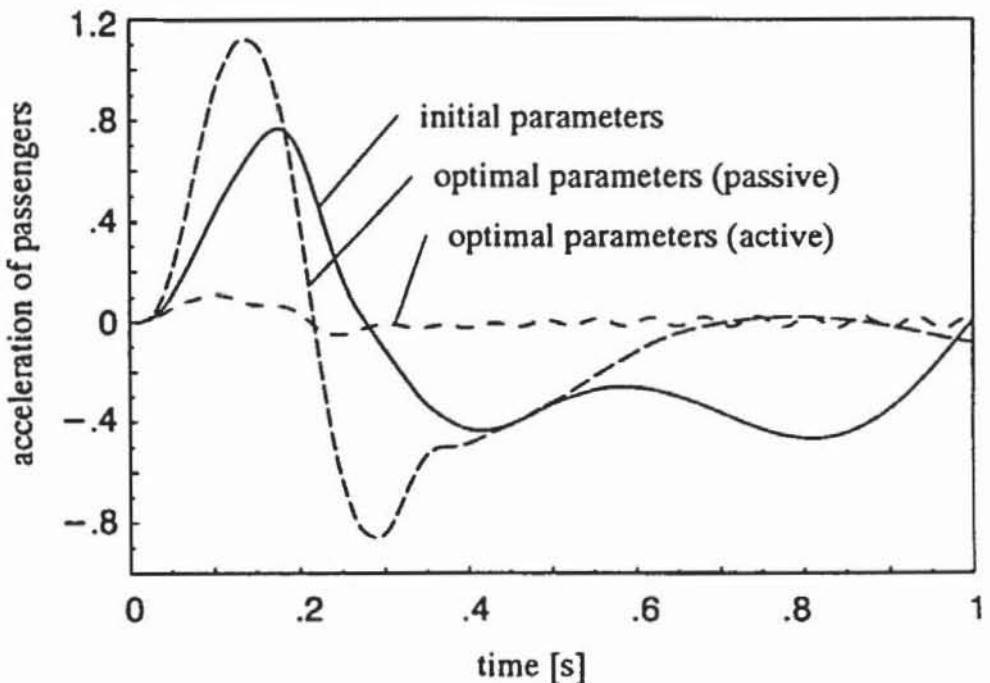


Figure 16: Optimization of ride comfort, progressive damper

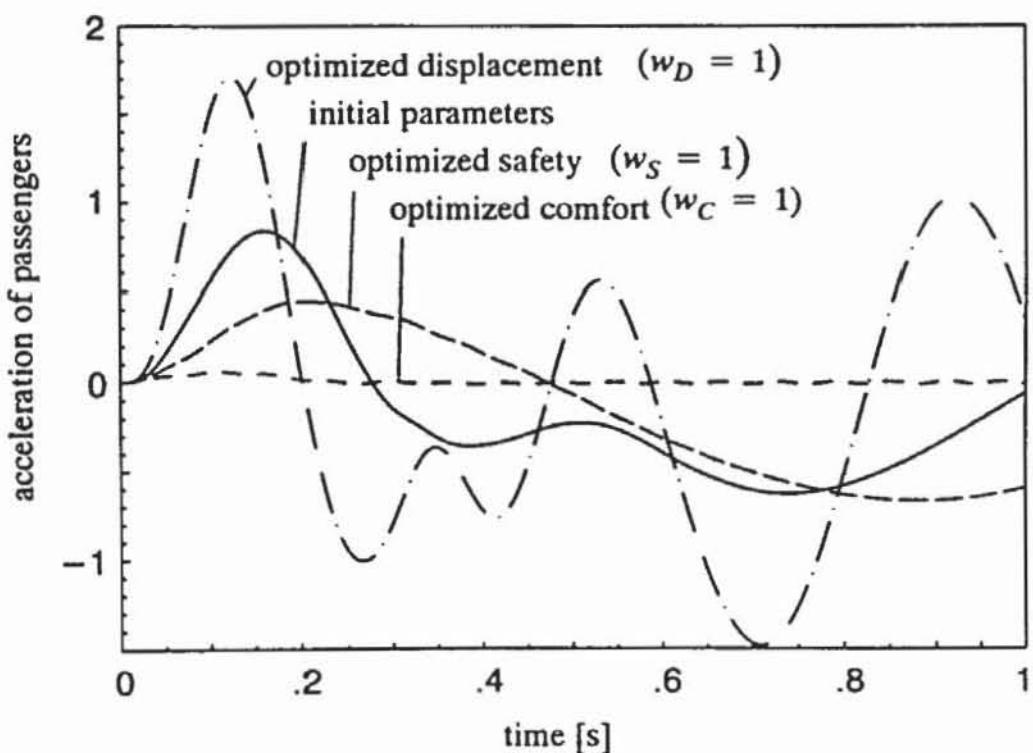


Figure 17: Optimization by single criteria, linear damper

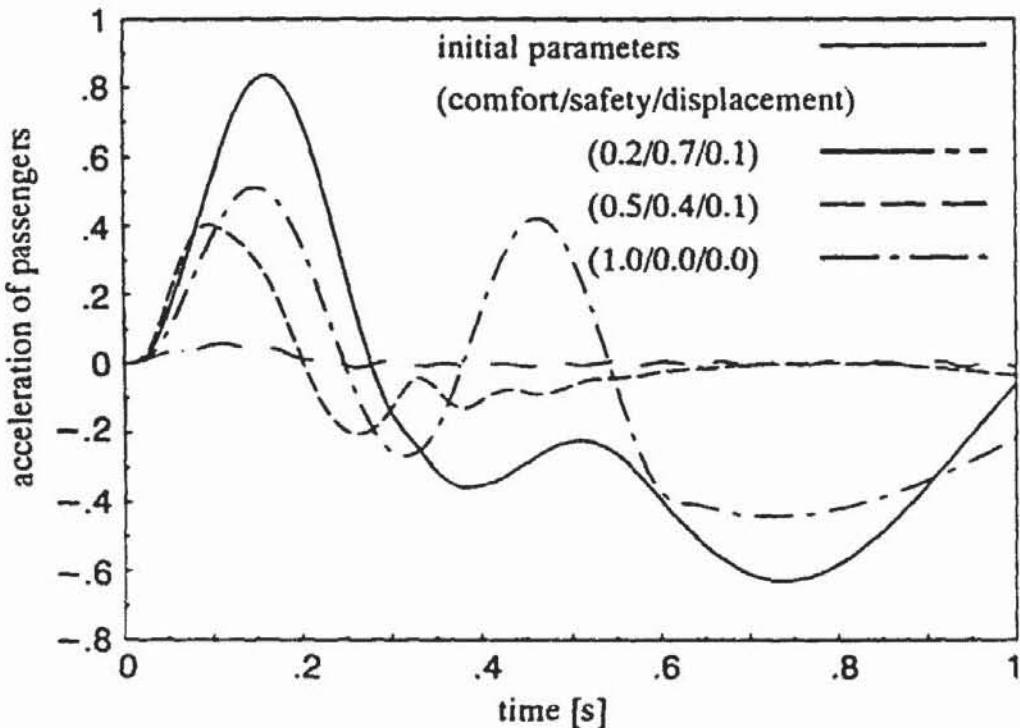


Figure 18: Optimization by mixed criteria, linear damper

11. Conclusion

In this paper an integrated modeling, simulation, visualization, and optimization of multibody system dynamics is introduced. A unified general data model including the graphic description is presented. To support the preceding CAD-3D-modeling stage, a unified spatial graphic representation for multibody elements is designed. Object-oriented classes and operations are then implemented in a system independent multibody modeling kernel library and integrated in a commercial CAD-3D system. From the multibody model data base, an integrated Newton-Euler formalism generates a set of symbolical ordinary differential equations, which are solved by explicit multistep integration algorithms. Thereby, a minimal set of generalized coordinates is specified during numerical integration without restart of the integration algorithm, using the projection method within the coordinate partitioning approach. The visualization of the crank slider mechanism demonstrates that this integrated approach fits the criteria of a modular, automated design and simulation environment. It is shown that symbolic computations are also very efficient in the dynamical analysis of flexible multibody systems. Further, the optimization of design parameters is introduced and the adjoint variable method is presented. For the gradient computation a combination of NEWEUL and MAPLE generated symbolical equations of motion and

of adjoint variables. The multicriteria optimization overcomes the conflicts typical for engineering problems. As an example the optimization of a vehicle's suspension system is presented.

References

- 1 Roberson, R.E., Schwertassek, R. (1988): *Dynamics of Multibody Systems*. Springer, Berlin.
- 2 Nikravesh, P.E. (1988): *Computer-aided Analysis of Mechanical Systems*. Prentice-Hall, New Jersey.
- 3 Haug, E.J. (1989): *Computer-aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, Boston.
- 4 Shabana, A. (1989): *Dynamics of Multibody Systems*. Wiley, New York.
- 5 Kortüm, W.; Schiehlen, W. (1985): General purpose vehicle system dynamics software based on multibody formalisms. *Vehicle System Dynamics* 14, 229-263.
- 6 Bianchi, G.; Schiehlen, W. (eds.) (1986): *Dynamics of Multibody Systems*. Springer-Verlag, Berlin.
- 7 Kortüm, W.; Sharp, R. S. (1991): A report on the state-of-affairs on application of multibody computer codes to vehicle system dynamics. *Vehicle System Dynamics* 20, 177-184.
- 8 Schiehlen, W. (ed) (1990): *Multibody Systems Handbook*. Springer-Verlag, Berlin.
- 9 Orlandea, N. (1973): *Node-Analogous Sparsity-Oriented Methods for Simulation of Mechanical Systems*. Ph.D. dissertation, University of Michigan.
- 10 Haug, E.J. (1989): *Computer Aided Kinematics and Dynamics of Mechanical Systems*. Allyn and Bacon, Boston.
- 11 Rosenthal, D.E., Sherman, M.A. (1986): High performance multibody simulation via symbolic equation manipulation and Kane's method. *Journal of Astronautical Sciences* 34, 223-239.
- 12 Kreuzer, E. (1979): *Symbolische Berechnung der Bewegungsgleichungen von Mehrkörpersystemen*, Ph.D. Dissertation, Stuttgart.

- 13 ARIES Conceptstation Software Simulation Mechanism Reference. Aries Technology Inc., Lowell, MA, 1990.
- 14 Hollar, M.G.; Rosenthal, D.E. (1991): *Concurrent Design and Analysis of Mechanisms*. Rasna Corporation, San Jose.
- 15 Otter, M.; Hocke, M.; Daberkow, A.; Leister, G. (1990): *Ein objektorientiertes Datenmodell zur Beschreibung von Mehrkörpersystemen unter Verwendung von RSYST*. Institut B für Mechanik, IB-16, Stuttgart.
- 16 Mortenson, M.E. (1985): *Geometric Modeling*. John Wiley, New York.
- 17 Daberkow, A. (1992): *Zur CAD-gestützten Modellierung von Mehrkörpersystemen*. Ph.D. Dissertation, Stuttgart.
- 18 Rühle, R. (1973): RSYST, ein integriertes Modulsystem mit Datenbasis zur automatischen Berechnung von Kernreaktoren. IKE 4-12 1973, Stuttgart.
- 19 SIGRAPH-CAD-3D. SIEMENS NIXDORF AG, München, 1992.
- 20 Otter, M.; Hocke, M.; Daberkow, A.; Leister, G. (1993): An object oriented data-model for multibody systems. In: *Advanced Multibody System Dynamics*. Kluwer, Dordrecht, 19-48.
- 21 Kreuzer, E.; Leister, G. (1991): Programmsystem NEWEUL'90, Anleitung AN-24, Institut B für Mechanik, Stuttgart.
- 22 Leister, G. (1991): Programmepaket NEWSIM. Anleitung AN-25, Institut B für Mechanik, Stuttgart.
- 23 Schiehlen, W. (1986): *Technische Dynamik*. Teubner Verlag, Stuttgart.
- 24 ACSL-Advanced Continuous Simulation Language Reference Manual. Inc. Concord/Mass.: Mitchell u. Gauthier Assoc., 1987.
- 25 Otter, M.; Gaus, N. (1991): ANDECS-DSSIM: Modular Dynamic Simulation With Database Integration. User's Guide, Version 2.1. Oberpfaffenhofen.
- 26 Leister, G. (1992): *Beschreibung und Simulation von Mehrkörpersystemen mit geschlossenen kinematischen Schleifen*. Ph.D. Dissertation, Stuttgart.
- 27 Blajer, W. (1991): Contribution to the projection method of obtaining equations of motion, *Mechanics Research Communications* 18, 293-301.

- 28 Baumgarte, J. (1972): Stabilization of constraints and integrals of motion in dynamical systems, *Computational Methods in Applied Mechanics and Engineering*, 1, 1-16.
- 29 Ostermeyer, G.-P. (1990): On Baumgarte stabilization for differential algebraic equations. In: *Real-Time Integration Methods for Mechanical System Simulation*, NATO ASI Series, Vol. F69, Springer-Verlag, Berlin-Heidelberg, 193-207.
- 30 Woernle, C. (1988): *Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbegrenzungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter*, Ph.D. Dissertation, Stuttgart.
- 31 Eppinger, M.; Kreuzer, E. (1990): Evaluation of methods for solving the inverse kinematics of manipulators, In: Proc. of the 8-th CISM-IFToMM Symp. on Theory and Practice on Robots and Manipulators, Cracow.
- 32 Blajer, W.; Schiehlen, W.; Schirm, W. (1993): Dynamic analysis of constrained multibody systems using inverse kinematics. *Mechanism and Machine* 28(3), 397-405.
- 33 Schiehlen, W.; Blajer, W. (1992): Closing conditions and reaction forces of multibody systems. *Zeitschrift für Angewandte Mathematik und Mechanik (ZAMM)* 72, T45-T47.
- 34 Schiehlen, W. (1991): Computational aspects in multibody system dynamics, *Computer Methods in Applied Mechanics and Engineering* 30, 569-582.
- 35 Rauh, J. (1988): *Ein Beitrag zur Modellierung elastischer Balkensysteme*, Ph.D. Dissertation, Stuttgart.
- 36 Bremer, H.; Pfeiffer, F. (1992): *Elastische Mehrkörpersysteme*, Teubner, Stuttgart.
- 37 Melzer, F. (1993): Symbolic computations in flexible multibody systems. In: Computer Aided Analysis of Rigid and Flexible Mechanical Systems (NATO ASI, Tróia, 27 June - 7 July 1993). Pereira, M.S. (ed.). To appear.
- 38 Sorge, K. (1992): *Mehrkörpersysteme mit starr-elastischen Subsystemen*. Ph.D. Thesis, München.
- 39 Bestle, D., Eberhard, P. and Schiehlen, W.: Optimizing of an actively controlled vehicle system. In: *Proceedings of the IUTAM Symposium Optimal Control of Mechanical Systems* (Moscow, Russia, 20-24 April 1992). Chernousko, F.L. (ed.). To appear.

- 40 Osyczka, A. (1984): *Multicriteria Optimization in Engineering*. Ellis Horwood, New York.
- 41 Fletcher, R. (1987): *Practical Methods of Optimization*. Wiley, Chichester.
- 42 Haug, E. J. (1987): Design Sensitivity Analysis of Dynamic Systems. In: *Computer Aided Design: Structural and Mechanical Systems*, Mota-Soares, C.A. (ed.), Springer, Berlin.
- 43 Bestle, D. and Eberhard, P. (1992): Analyzing and Optimizing Multibody Systems. *Mechanics of Structures and Machines* 20, 67-92.
- 44 Char, B.W. et. al. (1990): *MAPLE-Reference Manual*. Waterloo Maple Publ., Waterloo.