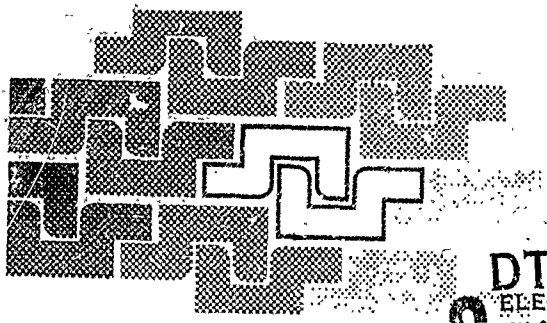


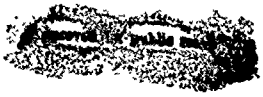
COMPUTER AIDED ANALYSIS OF RIGID AND FLEXIBLE MECHANICAL SYSTEMS

NATO-Advanced Study Institute

Volume II
Contributed Papers

Eds.: Manuel S. Pereira and Jorge A. C. Ambrósio

TRÓIA, PORTUGAL, 27 JUNE - 9 JULY, 1993



Proceedings of the
NATO-Advanced Study Institute
on

COMPUTER AIDED ANALYSIS OF RIGID AND FLEXIBLE MECHANICAL SYSTEMS

Volume II
Contributed Papers

Eds.: Manuel S. Pereira and Jorge A. C. Ambrósio

TRÓIA, PORTUGAL, 27 JUNE - 9 JULY, 1993

NATO-Advanced Study Institute

COMPUTER AIDED ANALYSIS OF RIGID AND FLEXIBLE MECHANICAL SYSTEMS

TRÓIA, PORTUGAL, 27 JUNE - 9 JULY, 1993

MAIN SPONSOR

NATO - North Atlantic Treaty Organization

OTHER SPONSORS

U.S. Army TARDEC - United States Army, TARDEC

NSF - National Science Foundation

JNICT - Junta Nacional de Investigação Científica e Tecnológica

FLAD - Fundação Luso Americana para o Desenvolvimento

OTHER SUPPORTS

HEWLLET PACKARD PORTUGAL

RANK XEROX PORTUGAL

SOGAPAL

LUSANOVA

DIRECTOR

MANUEL S. PEREIRA

ORGANIZING COMMITTEE

Michel Geradin, Professor, LTAS, Université de Liége, BELGIUM

Edward J. Haug, Professor, C. C. A. D., University of Iowa, USA

Manfred Hiller, Professor, University of Duisburg, Fachgebiet Mechatronic, Germany

Parviz Nikravesh, Professor, Aero & Mech. Eng. Dpt., Un. of Arizona, USA

Manuel S. Pereira, Ass. Prof., IDMEC/Instituto Superior Técnico, PORTUGAL

ORGANIZED BY

IDMEC - Instituto de Mecânica
Instituto Superior Técnico
Universidade Técnica de Lisboa

IDMEC, Av. Rovisco Pais, 1096 Lisboa Codex, PORTUGAL,
Ph.: 351-1-847 3437, Fax: 351-1-847 4045

CONTENTS

DYNAMICS OF CONSTRAINED SYSTEMS BASED ON MASS-ORTHOGONAL PROJECTIONS

M. Sofer, D. Bach & H. Brauchli
Institut für Mechanik - ETH Zentrum - CH-8092 Zürich

Abstract: We describe a method based on mass-orthogonal projections that can be used to eliminate the algebraic variables in the DAEs of index 2 of multibody system dynamics, reducing them to ODEs. A choice among the infinitely many ODE systems that describe the dynamics correctly is proposed. A reduction of the computational cost of linear algebra operations is described. We present DYNAMITE, a tool-kit based software for the simulation of multibody systems that implements the methods discussed, and illustrate its application to a low-voltage circuit breaker.

1. Introduction

Different approaches are possible for the description of the motion of a multibody system. A review of methods commonly in use may be found in [11] or [15]. A minimal description, e.g. with Lagrange's method, provides the smallest system of differential equations but suffers a few drawbacks. Some of these are: the impossibility of describing the whole configuration space in a single coordinate chart without singularities in the presence of kinematical loops; the difficulty inherent in the calculation of the eliminated constraint reactions and the consideration of friction forces; the fact that the equations may be extremely complicated, i.e., the functions describing the differential equations are difficult to derive and expensive to compute; the lack of modularity, in the sense that the knowledge of the equations of motion of subsystems is difficult to use in the derivation of the equations of the full system.

For these reasons one of the widely used procedures in the analysis of multibody systems is to describe the system redundantly: one uses more variables than the strictly necessary to describe the motion locally, and imposes constraints on the motion. With this additional expense it is possible to obviate all of the above disadvantages of Lagrange's method. The equations to be solved form initially a mixed algebraic-differential system of equations (DAEs), the algebraic variables being the constraint reactions. These can be treated directly as DAEs using different methods [1,7,8,10,12,13], or they can be reduced to a set of ordinary differential equations (ODEs) by eliminating the algebraic variables.

We describe here a method based on mass-orthogonal projections developped in the past few years [3,5,6,18-21]. The constraint reactions are eliminated in an extremely general manner by the usage of mass-orthogonal projectors (α,β) in velocity space; α projects onto the subspace of admissible velocities, β is its complement. In the case of workfree constraints the projectors can be used to split the DAEs into a pure ODE for the motion of the constrained system and algebraic expressions for the constraint reactions in terms of position and velocity. The resulting ODEs for the motion are not uniquely determined by

the projectors, we describe in section 2 two infinite families of ODEs that correctly describe the motion of the constrained system and discuss briefly some sample ODE formulations contained in these families.

In section 3 we present an organisation of the linear algebra computations that, under certain assumptions, can significantly reduce the computational cost of the right hand side of the ODEs. These assumptions can be met in a redundant description of the multibody system in which each of the bodies is modelled separately.

The present method is well suited for numerical simulations of multibody systems. In section 4 we describe DYNAMITE [3], a tool-kit based software that implements the methods and algorithms discussed here. The simulation with DYNAMITE of a low-voltage circuit breaker [3] is illustrated in section 5. Kinematical loops, friction in the joints and collisions are the main difficulties presented by this industrial application.

The equations obtained for the description of constrained systems are not only of value as a basis for simulation, but they lend themself for analytical developments. It is possible to describe holonomically constrained conservative systems with hamiltonian ODEs, thus allowing the usage of the canonical instrumentarium [20]. The configuration manifold of a holonomically constrained system can also be viewed as imbedded in a euclidean space with constant metric. Simple formulas for the curvature of the configuration manifold of the constrained system may be obtained [6]. According to Synge [22] this has implications for the stability of motion for orbits of the constrained system. Corresponding results will be presented elsewhere.

2. Mass-orthogonal projectors and ODE formulations

Consider the motion of a dynamical system described with n coordinates q, $r \le n$ non-holonomic velocity parameters u and mass matrix M; let A be the inverse of M and T(q,u) be the expression for the kinetic energy.

Let the system be subject to m constraints described by the (rxm)-matrix E as

$$E^T u = 0 \ . \tag{1}$$

Remark that holonomic constraints have to be differentiated once to take this form. We consider here only scleronomic constraints and assume that the matrix E has full row rank, i.e., that the constraints are independent.

Introducing the canonical momentum $p = Mu$, the motion of the system under the applied generalised forces Q is described by the DAE system of index 2

$$\begin{aligned} u &= A\,p \ , & \dot{q} &= \bar{E}\,u \ , \\ \dot{p} &= \dot{Q} = \bar{Q} + E\,\lambda \ , & E^T u &= 0 \ . \end{aligned} \tag{2}$$

In this expression $\bar{Q}$ has to be calculated as

$$\bar{Q} = Q + \frac{\delta T}{\delta q} - C(u,u), \tag{3}$$

i.e., as the sum of the applied forces Q and additional inertia terms, quadratic in the velocities and due to the non-constant mass matrix and the non-holonomic velocity parameters [5].

We assume for the purposes of the present description that the generalised forces Q may depend on (t,q,u), but not directly on λ. This rules out friction forces; see [3] for their inclusion in the considerations.

2.1 Mass-orthogonal projectors

Consider the ortho-projector (in the metric defined by the mass matrix M) α on the null space of E; this projector appears naturally in the dynamics of the system in one form or the other, some other authors also use it explicitly [2,12,16,17]. Let β be its complementary projector. They satisfy the relations

$$\alpha\,\beta = 0\ ,\ \alpha^T M = M\alpha\ ,\ \beta^T M = M\beta\ ,\ \alpha\,\alpha = \beta\ ,\ \alpha\,\beta = \alpha\ . \tag{4}$$

The projectors can be calculated under our assumptions as

$$G = E^T A^{-1} E\ ,\ \beta = A^{-1} E\, G^{-1} E^T\ ,\ \alpha = I - \beta\,. \tag{5}$$

Introducing the matrices

$$\Omega = E^T G^{-1} \dot{E}\ ,\ \omega = \Omega\,\alpha + \alpha^T \Omega^T\ , \tag{6}$$

the derivatives of the projectors are given by

$$\dot{\alpha} = -\,\dot{\beta} = A^{-1}(\alpha^T \dot{M}\,\beta - \omega). \tag{7}$$

Let us introduce as a shorthand

$$x_\alpha = \alpha^T x\ ,\ x_\beta = \beta^T x \tag{8}$$

to denote the α resp. β projections of a co-vector (forces, momenta) x.

2.2 Formulation as ordinary differential equations (ODE)

There are many different systems of ordinary differential equations that describe the system (2) (see [18,21]), in the sense that their solution sets contain all trajectories (q(t),p(t)) of the DAE (2). A first infinite family of such ODE formulations is given by

$$\text{F1:}\qquad u = A\,p_\alpha + X\,p_\beta\ ,\qquad \dot{q} = B\,u\ , \tag{9}$$

$$\dot{p} = Q_\alpha + \dot{\alpha}^T p_\alpha + Y\,p_\beta\ ,$$

where the matrices X and Y can take any value, depending (regularly) on any of the variables. The solutions of (2) constitute an invariant manifold ($p_\beta = 0$) for these equations. Remark that these equations represent a second order system in q if and only if the matrix $A\alpha^T + X\beta^T$ is invertible. If one chooses $X = 0$ then every trajectory of (9) satisfies the constraints (1), but in general is a solution of the DAE (2) only if it lies on the invariant manifold ($p_\beta = 0$).

A second infinite family of ODE formulations of (2) is given by

$$\text{F2:}\qquad u = A\,p_\alpha\ ,\qquad \dot{q} = B\,u\ , \tag{10}$$

$$\dot{p} = Q_\alpha - \dot{\alpha}^T p_\beta + \beta^T x\ ,$$

where the co-vector x can take any value, depending (regularly) on any of the variables. All solutions (q(t),p(t)) of (10) are trajectories of the constrained system, in the sense that there is a λ such that (q,p_α,λ) solves (2).

Formulations in the first family have the advantage that, for trajectories that satisfy ($p_\beta = 0$), the variable p retains its physical interpretation as the canonical momentum of the free system. Comparing (9) and (2), this implies that the non-working generalised reactions are given by $E\lambda = \dot{p} - Q$ and are calculated during the evaluations of the right hand side of the differential equation. The reactions are therefore available during any numerical integration of system (9).

In the second family of formulations, the variable p cannot in general be interpreted as the canonical momentum of the free system; it is just an auxiliary variable. In the general case the constraint reactions are not immediately available during a simulation. An

advantage of the second family is the fact that any solution of the differential equations (10) corresponds to a trajectory of the constrained system.

The intersection of both families of formulations is not empty; indeed, it follows easily from (9) and (10) that it is given by

$$\text{F3:} \qquad u = A\,p_\alpha\,, \qquad \dot{q} = \bar{B}\,u\,, \tag{11}$$
$$\dot{p} = \bar{Q}_\alpha + \alpha^T(\alpha^T\beta^T)\,p + \beta^T Z\,p_\beta\,,$$

where Z is any matrix depending (regularly) on any of the variables.

2.3 Examples of ODE formulations

We chose as examples the formulations described in Table 1. Remark that these formulae apply only in the case where M (and hence A) are constant; the generalisation is essentially trivial but leads to bulkier formulae. Formulation 1 is the standard DAE formulation of index 1 solved for the kinematical variables, as described for instance in [10]. Formulations 1 to 4 were already described and discussed in [18], whereas formulation 5 corresponds to the "projected equations of motion" presented in [16,17]. The term $(-\mu p_\beta)$ in formulation 6 causes the invariant manifold ($p_\beta = 0$) to be asymptotically stable.

Formulation	Fam.	u	$\dot{p}$
1	F1	$A\,p$	$\bar{Q}_\alpha - \Omega\,A\,p$
2	F1	$A\,p_\alpha$	$\bar{Q}_\alpha - \Omega\,A\,p_\alpha$
3	F2	$A\,p_\alpha$	$\bar{Q} + \Omega^T A\,p$
4	F3	$A\,p_\alpha$	$\bar{Q}_\alpha + \alpha^T\Omega^T A\,p - \Omega\,A\,p_\alpha$
5	F3	$A\,p_\alpha$	$\bar{Q}_\alpha + \Omega^T A\,p - \Omega\,A\,p_\alpha$
6	F3	$A\,p_\alpha$	$\bar{Q}_\alpha + \alpha^T\Omega^T A\,p - \Omega\,A\,p_\alpha - \mu\,p_\beta$

Table 1: Examples of ODE Formulations (constant M)

All formulations describe the constrained dynamics properly, but they differ in some properties that may be relevant for their numerical treatment. See [18,19] for some numerical comparisons and [21] for a thorough discussion.

Table 2 shows the integrals of motion for the ODE of each of the selected formulations. The integrals of formulation 1 are actually a disadvantage, as they cause the well known numerical drift of the solutions. The second column corresponds to the norm of p_β in the metric of A (which is the correct metric for co-vectors); the * for formulation 6 indicates that this norm diminishes exponentially (with exponent -2μ). In the third column appear the possible holonomic constraints. Finally, the last column corresponds to the

Formulation	$E^T u$	$\|p_\beta\|_A$	Holonomic Constraints	Energy equation
1	m			
2	(≡ 0)	1	≤m	1
3	(≡ 0)		≤m	1
4	(≡ 0)	1	≤m	1
5	(≡ 0)		≤m	1
6	(≡ 0)		≤m	1

Table 2: Integrals of motion

energy equation, which gives a proper integral of motion only when the applied forces Q are conservative. Remark that the energy equation is satisfied in a stronger sense for formulation 4. Formulations 4 and 6 have the further property that a solution on the invariant manifold ($p_\beta = 0$) is stable if and only if it is stable as a trajectory of the DAE (2), as can easily be shown using the properties of the second family and the non-increasing nature of $\|p_\beta\|$.

The constraints (1) can be written equivalently with any invertible (mxm)-matrix S as

$$S E^T u = 0 \,. \tag{12}$$

Formulations 2, 4 and 6 are invariant with respect to the choice of S; 1, 3 and 5 are not. Formulations 3 and 5 have invariant manifolds corresponding to possible holonomic constraints; the motion on these manifolds, however, depends on the choice of S, i.e., on the formulation of the constraints.

We have chosen formulation 4 for our further developments. This choice is based on our numerical experiments [18,19], on the invariance and stability properties of the equations, and on the fact that there is no risk of artificially stiffening the equations, as may be the case for formulation 6.

3. Numerical linear algebra

The computation of the matrices appearing in the formulae above is not necessary. An efficient computation avoids both the matrix inversion and matrix multiplications, replacing them with the solution of systems of equations and matrix-vector multiplications. Apart from the calculation of inertia terms and applied forces, the bulk of the computational effort in all formulations is constituted by the computation of the Cholesky factor of the matrix G. Once the factor is computed, formulations 1 and 3 require the solution of a single equation with matrix G, the others require two. For some systems it may be advisable to organise the computations in a different manner, solving instead equations with the indefinite system matrix

$$\begin{bmatrix} M & E \\ E^T & 0 \end{bmatrix}. \tag{13}$$

The calculations presented above in terms of G amount to the inversion of (13) (using range space techniques) in the form

$$\begin{bmatrix} M & E \\ E^T & 0 \end{bmatrix}^{-1} = \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & AE \\ 0 & -I \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & G^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ E^T A & -I \end{bmatrix} \tag{14}$$

where I is the (rxr) identity matrix. Remark that solving an equation with matrix (13) reduces then to solving an equation with the smaller and positive definite matrix G, and a few matrix-vector multiplications.

For every multibody system there is a description with a *constant diagonal mass matrix* M such that *each body is modelled separately*, as detailed in the next section. This description leads to large systems of equations when compared to alternative descriptions (e.g. describing a spanning tree of the system in minimal coordinates), but has other advantages (see next section). We describe here how to use the special structure of the equations to reduce significantly the computational cost of the linear algebra calculations.

First note that, thanks to the constant diagonal M and A, the inversion according to equation (14) becomes much cheaper. Indeed, $A = M^{-1}$ need be calculated only once at the beginning of a simulation, and the multiplication with A is extremely inexpensive.

For most mechanical systems each constraint involves at most two different bodies. This is expressed in the sparsity (with constant sparsity pattern) of the matrix E in any description where the bodies are modelled separately. Indeed, in each column of E only the rows corresponding to velocity parameters of the concerned bodies are occupied, i.e., at most a fraction 2/(number of bodies) of the rows. The multiplication of a vector with either E or its transpose can be computed efficiently taking advantage of this sparsity.

Furthermore, the sparsity can be used to reduce the cost of the Cholesky factorisation of the matrix $G = E^T A E = R^T R$. The constraints – i.e. the columns of E – are reordered once at the beginning of the simulation to obtain a sparse factor R. This reordering can be done using any standard algorithm, for instance minimal degree [9] or lexicographic [14] orderings. At each evaluation of the right hand side in the ODE one has then to proceed to compute R using the known sparsity structure; this can be done either with a sparse Cholesky algorithm applied to G, or else by computing a sparse QR-decomposition of the matrix $\sqrt{A}E$ (for instance with a sequential row orthogonalisation method [4]). In the second case, the re-ordering of the rows of E to reduce the computational effort has to be done only once at the beginning of the simulation.

Already in the relatively small seven-body mechanism described in [15], where in our description E and R are occupied to 15% respectively 37%, the consideration of the sparsity reduces the factorisation time by a factor of about four.

By a careful exploitation of the sparsity structure it is to be expected that the computational effort will grow linearly in the number of constraints and bodies for 'typical' systems (the factor R can be shown to be full in any system where one body is connected to every other body, so that the sparse factorisation may not be advantageous for such systems). A direct treatment of matrix (13) as a sparse matrix has the disadvantage that the pivot choice computed initially has to be monitored during the factorisation to insure the stability of the algorithm, as the matrix is indefinite. Recursive elimination [13], a linear-growth algorithm, is applicable to systems with tree structure; loop-closing constraints have to be treated separately in other systems.

4. DYNAMITE

4.1 Basic features

DYNAMITE [3] is an implementation of the projection method for the description of multibody systems as described above, in particular of formulation 4. To meet industrial needs, this basic formulation has been extended to problems with time-dependent constraints, with sliding friction dependent on non-working constraint forces and also to

problems of inverse kinetics. The additional algebraic equations for the treatment of sliding friction are minimal in number and of index 1 but non linear, while the additional algebraic equations caused by inverse kinetics are linear and of index 1 even when the imposed motion is strictly satisfied at the velocity level. The time-dependent constraints are taken into account in the projections, thus avoiding additional equations.

DYNAMITE treats each body separately and uses absolute coordinates: cartesian coordinates of the center of gravity and rotation angle for planar systems, and cartesian coordinates of the center of gravity and two unit vectors in principal directions of inertia for spatial systems. The derivatives of the coordinates (for planar systems), the velocity of the center of gravity in components of the inertial system and the rotational velocity in components of the body fixed system (for spatial systems) are used as velocity parameters. This system description satisfies the conditions of section 3, i.e., separate treatment of each body and constant and diagonal mass matrices. In addition to that, there are other advantages of the chosen description. Thanks to the separate treatment of each body using absolute coordinates, a kinematical analysis prior to simulation is not necessary in the case of statically determined systems; the formulation of constraints and forces and the calculation of inertia terms are simple and inexpensive; subsystems are easily appended, deleted or modified. Furthermore, the coordinates are globally defined so that coordinate singularities are avoided.

The class of systems that can be simulated with DYNAMITE is currently restricted to statically determined systems.

DYNAMITE consists of three main parts:

- *Tool-kits* (one for planar systems, one for spatial systems): bodies, constraints, forces. The tool-kits create numerical matrices and vectors to describe the system (A, E, C, Q etc.). New constraints or forces can easily be appended to the tool-kits by the user.
- *Linear algebra processor*: Fully numerical, system independent calculation of the equations of motion. The linear algebra processor works with the system matrices and vectors created by the tool-kits. Special attention has been paid to the modularity to simplify maintenance and generalization.
- *Integrators*: LSODAR and DASRT

DYNAMITE is a joint developement of the Swiss Federal Institute of Technology Zurich and Oerlikon-Contraves Zurich; it is sponsored by the Commission for the Promotion of Scientific Research of the Swiss Federal Departement of Economy.

4.2 Future Developements

DYNAMITE will be extended to perform parameter-sensitivity analysis (with analytically generated sensitivity matrices) and to handle small and large elastic deformations. Other possible extensions are: numerical methods for statically indetermined systems, adequate methods for stiff systems.

4.3 Previous Industrial Applications

DYNAMITE has been tested on theoretical systems and on the two standard reference examples in [15]. It has also been applied successfully to industrial problems:

- simulation and calculation of energy consumption of a Gatling gun (Oerlikon-Contraves, Zürich)

* simulation of a low voltage circuit breaker
(in collaboration with ABB Corporate Research, CH-Baden-Dättwil; see next section)

5. Simulation of a Low Voltage Circuit Breaker with DYNAMITE

5.1 System description

The mechanism, depicted in Figure 1 in the position "closed", consists of six parts (Nr. 1 to 6). They are interconnected by four revolute joints (D connects 3 and 4, E connects 1 and 3 and F connects 4, 5 and 6). Three revolute joints connect parts of the mechanism to the inertial system (A with 1, B with 2 and C with 5).

Figure 1: Circuit breaker in the closed configuration

Attached to body 2 is a manipulation handle to bring the circuit breaker from the position "open" to the position "closed". Body 6 is the transmission part between the circuit contact f and the wire attached at the other end of this body.

Regions with unilateral constraints (where impacts can occur depending on the parameter values) are marked in fig. 1 by a, b, c, d, e (G is a system-fixed bolt; impacts with G can occur together with bodies 2 and 3) and f. Regions g (bolt H is part of body 3) and h (where a hook holds body 1) are other regions of unilateral constraints. The unilateral constraints e (body 2), f and g are active when the circuit is closed and a, b and e (body 3) are active when the circuit is broken.

The mechanism has three degrees of freedom when no unilateral constraint is active: the angles of bodies 1, 5 and 6 define fully the configuration of the mechanism.

The mechanism contains two springs: one between bodies 5 and 6 and the other between body 2 (point I in Figure 1) and the bolt of revolute joint D, the latter being the main tension spring responsible for the the motion of the mechanism.

The mechanism stays in the closed position as long as body 1 is held in h by a hook. Taking away that hook starts the transition from "closed" to "open"; this process takes less then 0.04 second. What happens during this transition can be described roughly as follows: The force of the elongated main tension spring acting on joint D leads via body 3 and joint E to a torque acting counterclockwise on body 1. Body 2 stays in its initial position as long as the line of application of the main tension spring is on the left hand side of joint B. Body 3 turns clockwise until the line of application of the main tension spring changes from the left hand side of joint E to the right hand side; it then turns counterclockwise. The final position is defined by the activity of unilateral joints a, b and e (body 3), as explained above.

5.2 Simulation and Experiment

Figures 2 and 3 show the calculated angles of bodies 5 and 6 defining together the position of body 6 (which is functionally relevant). Figure 2 shows also the experimentally determined time history of the angle of body 5.

Figure 2: Angle of body 5 (ψ), experiment and simulation

Figure 3: Angle of body 6 (φ), simulation

The simulation is based on parameter values determined by experiments. Parameter variations and variations in the modelling of damping, friction and impacts are important when studying reliability of the calculations. Thereby, the not negligible variation of the mentioned phenomena in reality as well as the complexity of the mechanism - body 6 has no fixed hinge - raise special difficulties.

The parameter fitting showed the importance of sensitivity analysis which will be the next extension of DYNAMITE. Thanks to its features - especially the matrix-structure of the equations - DYNAMITE is well-suited for this extension. In the developement cycle, where minimizing prototyping through reliable calculations is a main objective, sensitivity analysis plays an even more important role than in the analysis of existing mechanisms.

Acknowledgement

The authors gratefully acknowledge the support of the KWF (Commission for the Promotion of Scientific Research of the Swiss Federal Departement of Economy).

References

[1] O. P. Agrawal & S. Saigal, Dynamic Analysis of Multi-Body Systems Using Tangent Coordinates, Comp. Struct. 31 (1989), 349-355.

[2] T. Alishenas, Zur numerischen Behandlung, Stabilisierung durch Projektion und Modellierung mechanischer Systeme mit Nebenbedingungen und Invarianten, Thesis TRITA-NA-9202, Royal Inst. Techn. Stockholm, March 1992.

[3] D. Bach, H. Brauchli & O. Melliger, DYNAMITE: Multibody Dynamics via Projection Method, Proceedings of the International Symposium on Advanced Multibody Dynamics, Stuttgart, March '93, Kluwer Academic Publishers, Dordrecht, 1993.

[4] A. Björck, Least Square Methods, Handbook of Numerical Analysis I, Eds. P. G. Ciarlet & J. L. Lions, North-Holland, 1990.

[5] H. Brauchli, Mass-Orthogonal Formulation of Equations of Motion for Multibody Systems, J. Appl. Math. Phys. (ZAMP), 42 (1991) 169-182.

[6] H. Brauchli & R. Weber, Dynamical equations in natural coordinates, Comp. Meth. Appl. Mech. Eng. 91 (1991) 1403-1414.

[7] C. Führer & B. Leimkuhler, Formulation and Numerical Solution of the Equations of Constrained Mechanical Motion, Forschungsbericht DFVLR, Oberpfaffenhofen (1989).

[8] C. W. Gear, G. K. Gupta & B. J. Leimkuhler, Automatic Integration of the Euler-Lagrange Equations with Constraints, J. Comp. Appl. Math. 12 (1985).

[9] A. George & J. W. H. Liu, The Evolution of the Minimum Degree Ordering Algorithm, SIAM Review 31 (1989), 1-19.

[10] E. Hairer & G. Wanner, Solving Ordinary Differential Equations II, Springer Verlag, Berlin 1991.

[11] R. L. Huston, Multibody dynamics - modeling and analysis methods, Appl. Mech. Rev. 44 (1991) 109-117.

[12] Ch. Lubich, Extrapolation Integrators for Constrained Multibody Systems, IMPACT Comp. Sci. Eng., 3 (1991), 213-234.

[13] Ch. Lubich, U. Nowak, U. Pöhle & Ch. Engstler, MEXX - Numerical Software for Integration of Constrained Mechanical Multibody Systems, preliminary notes.

[14] D. J. Rose, R. E. Tarjan & G. S. Lueker, Algorithmic Aspects of Vertex Elimination on Graphs, SIAM J. Comput. 5 (1976), 266-283.

[15] W. Schiehlen (ed.), Multibody Systems Handbook, Springer, Berlin 1990.

[16] B. Simeon, Numerical Integration of Multibody Systems by a Projection Technique, Technical Report TUM-M9201, Math. Institut T.U. München, 1992.

[17] B. Simeon, An Extended Descriptor Form for the Numerical Integration of Multibody Systems, J. Appl. Numer. Math., Special Issue on the Proceedings of the NUMDIFF 6 Conference, Halle, September 1992.

[18] M. Sofer, O. Melliger & H. Brauchli, Numerical Behaviour of Different Formulations for Multibody Systems, Proc. of the First European Conf. on Num. Methods in Eng., Elsevier Science Publishers B.V., Brussels 1992.

[19] M. Sofer, O. Melliger & H. Brauchli, ODE formulations for multibody dynamics: numerical aspects, Proceedings of the International Symposium on Advanced Multibody Dynamics, Stuttgart, March '93, Kluwer Academic Publishers, Dordrecht, 1993.

[20] M. Sofer & H. Brauchli, Hamiltonian Description of Holonomically Constrained Multibody Systems (submitted for publication).

[21] M. Sofer & H. Brauchli, ODE Methods for Constrained Dynamics (in preparation).

[22] J. L. Synge, On the geometry of dynamics, Phil. Trans. A 226 (1927), 31-106.

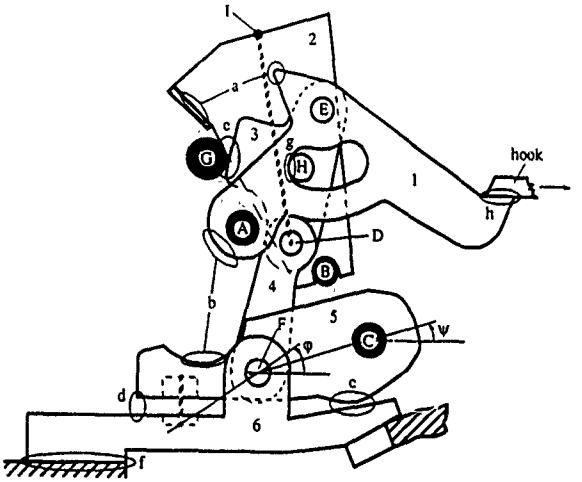


Figure 1: Circuit breaker in the closed configuration

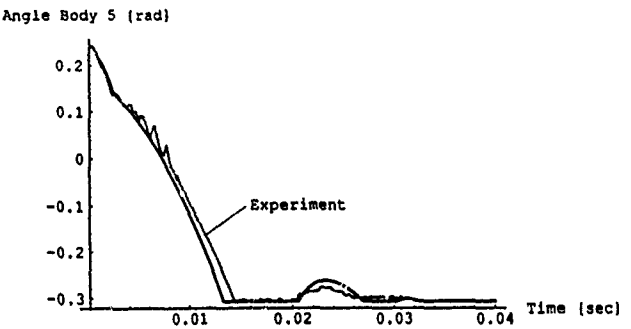


Figure 2: Angle of body 5 (ψ), experiment and simulation

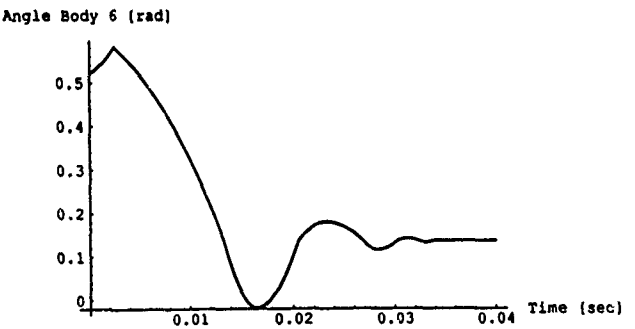


Figure 3: Angle of body 6 (φ), simulation

More Than Redundant Coordinates Formulation in Rigid Multibody Dynamics

M. Borri. C.G. Franchi. P. Mantegazza

Politecnico di Milano
Dipartimento di Ingegneria Aerospaziale
via Golgi 40. 20133 Milano. Italy

Abstract

The dynamics of rigid multibodies is traditionally formulated by means of either minimal or redundant coordinates methods. An alternative approach is here proposed whereby a set of more than redundant coordinates is adopted. As a result, the equations defining the constraints and the motion of the bodies are decoupled, many physically meaningful response variables are directly available while the constraint conditions are imposed in a very natural way. The first part of the paper presents the basic concepts and the theoretical developments of the formulation while the second part develops a numerical approximation for the methodology proposed in the first part. The nonlinear system of differential-algebraic equations ruling the motion of the system is written in its weak form and solved by applying a Newton-Raphson procedure to a finite elements approximation in the time domain. The details of the numerical implementation of this method will be discussed and a solution procedure will be presented. Some numerical examples involving tree and closed loop topologies will finally prove the capability of the present formulation in handling multibody dynamics.

1 Introduction

The definition of general purpose algorithms for the analysis of the dynamics of rigid multibodies represents an interesting task in modern computational mechanics. The literature on this subject is very rich, filling any aspect of the problem [1,2,3,4]. Various approaches have been proposed to analyse the motion of a moltibody system. Starting from a common set of primitive equations, the mechanical system can be described in its state space by means of a set of Ordinary Differential Equations (ODE) or in its physical space by means of a set of Differential Algebraic Equations (DAE) [5]. In the first family one finds the minimal set methods [6,7,8] and in the second the redundant coordinates methods [9,10,11,12].

This paper presents a new formulation to face the dynamics of rigid multibodies starting from its basic concepts and trying to explain its most innovative and interesting features. In the development of this work we try to take in some particularly felt requirements in multibody dynamics. At first we chose to have at disposal a wide set of primary unknowns with an immediate and univocal physical meaning allowing to avoid cumbersome and mathematically sophisticated pre and post processings to prepare the data and to recover interesting design parameters [13,14,15,16]. Moreover we organise the multibody model in such a way to solve many different problems such as direct and inverse kinematic and dynamic analysis, control system design and stability analysis in a unified framework. We rearrange the equations of motion to obtain a significant decoupling between the bodies and the constraints, thus allowing an easy exploitation of the massive parallelism of the modern hardware and software architectures [17,18].

To this end we develope a general kinematic analysis to stress the relation between the absolute coordinates, used to define the position of the bodies, and the relative coordinates, used to define the reciprocal position of two constrained bodies. In this formulation the absolute and relative coordinates are considered independent parameters constrained by a kinematic equation. A set of new unknown vectors, called multipliers, is then defined. They are related to the internal forces of the constraints as the momenta are related to the external forces acting on a free body. Two interesting results can be obtained in this way. The multipliers permit to properly define the modified momenta and to write the equations of motion of a constrained rigid body as those of a free body. Moreover, the aim of decoupling the equations of the bodies and constraints is reached, the multipliers representing the interface between the elements of the multibody system. The decoupling is shown also by a variational principle ruling the motion of a multibody system in Lagrangian form. The availability of the relative coordinates and of the internal forces allows a natural and easy imposition of the

constraints and a direct evaluation of these two parameters that are basics in the design of control systems.

2 Kinematic Analysis

The kinematic model of the elementary multibody system presented in Figure 1 and consisting in two constrained rigid bodies will be here outlined and analysed. The origin of an Inertial Reference Frame (IRF) is placed at the point O, those of two Embedded Reference Frames (ERF) at the points P_1 and P_2 and the points of the bodies where the constraints are applied will be called Q_1 and Q_2. A set of Lagrangian coordinates for the rigid bodies can be obtained by joining the position vectors of the reference points P_1 and P_2 and the finite rotation vectors ρ_1 and ρ_2 that rotate the bodies with respect to a reference configuration. They can be summarized by using the enlarged vectors (e.v.) [31,32] notation as:

$$q^{\cdot}_{P_i} = (a_i, \rho_i) \qquad i = 1,2$$

A natural set of relative coordinates is formed by the distance vector e which connects the constrained points Q_1 and Q_2 and by a properly defined relative rotation ρ.

$$q^{R} = (e, \rho)$$

The absolute and relative position e.v are represented with a notation that will be maintained for some other enlarged vectors. The symbol at the right upper corner represents either the body to which the vector is referred, if it is a number, or a relative parameter if it is R. The symbol at the right lower corner represents the point to which the enlarged vector is referred. Two equations can be stated to relate the relative coordinates to the bodies:

$$b_2 = b_1 + e \qquad R(\rho_2) = R(\rho) R(\rho_1) \tag{1}$$

in which $R(\rho_1)$, $R(\rho_2)$ and $R(\rho)$ are the rotation matrices associated to the rotation vectors ρ_1, ρ_2 and ρ. It must be noted that the use of the finite rotation vector to parametrize the rotation group $SO(3)$ is not the only possibility and several alternatives have been proposed [19] The relative linear velocity measured at the points Q_1 and Q_2 can be defined by:

$$v^{1}_{Q1} = v^{2}_{Q1} + v^{R}_{Q1} \qquad v^{2}_{Q2} = v^{1}_{Q2} + v^{R}_{Q2} \tag{2}$$

manipulating which it is not difficult to prove:

$$v^{R}_{Q1} = \frac{d^{\textcircled{i}}}{dt} e \qquad v^{R}_{Q2} = \frac{d^{\textcircled{i}}}{dt} e \tag{3}$$

In the above equation and in the followings the symbol $\textcircled{i}$ represents the relative time derivative or variation, i.e. as evaluated by an observer embedded in the ith body.
The definition of a relative angular velocity is less problematic than that of the linear one because of its independence from the choice of the reference point It can thus be consistently defined as.

$$\omega^{R} = \omega^{2} - \omega^{1} \tag{4}$$

which can be manipulated to obtain:

$$\omega^R = \Gamma(\rho)\frac{d^{\Phi}}{dt}\rho = \Gamma^T(\rho)\frac{d^{\Phi}}{dt}\rho$$

Some parametrizations of the matrix Γ are presented in References 20 and 21. A relative velocity *e.v.* can be defined by stacking the above definitions:

$$w^R_{Q2} = \left(v^R_{Q2}, \omega^R\right) = \left(\frac{d^{\Phi}}{dt}e, \Gamma(\rho)\frac{d^{\Phi}}{dt}\rho\right) = X\left(q^R_{Q2}\right)\frac{d^{\Phi}}{dt}q^R_{Q2}$$

$$w^R_{Q1} = \left(v^R_{Q1}, \omega^R\right) = \left(\frac{d^{\Phi}}{dt}e, \Gamma^T(\rho)\frac{d^{\Phi}}{dt}\rho\right) = X^T\left(q^R_{Q2}\right)\frac{d^{\Phi}}{dt}q^R_{Q2}$$

These two vectors can be related by means of the transport operator [31 32] E associated to the relative displacement e:

$$w^R_{Q2} = E^T w^R_{Q1}$$

Moreover, indicating by w^1_{Q1} and w^2_{Q2} the velocity *e.v.* of the bodies:

$$w^1_{Q1} = \left(v^1_{Q1}, \omega^1\right) \qquad w^2_{Q2} = \left(v^2_{Q2}, \omega^2\right)$$

the kinematic equation for the constraint can be stated by:

$$w^2_{Q2} - E^T w^1_{Q1} - w^R_{Q2} = 0 \tag{5}$$

3 Differential Formulation

The equations of motion of each constrained rigid body can be written using Euler equations:

$$\dot{p}^i_{Q_i} + S_1(w^i_{Q_i})p^i_{Q_i} - Q^i_{Q_i} - Q^R_{Q_i} = 0 \tag{6}$$

In which p represents the momenta *e.v.*, i.e., the vector formed by the linear and angular momentum of each body, w is the velocity *e.v.* formed by the linear and angular velocity, Q and Q^R are the external and internal forces *e.v.*, i.e., the vectors formed by the resultant force and torque.
The internal forces *e.v.* on the two bodies are related by the equation:

$$Q^R_{Q1} + EQ^R_{Q2} = 0 \tag{7}$$

The vectors Q^R appear explicitly in the Euler equations to couple the equations of the bodies. Some elements can be defined to allow the decoupling of the Euler equations. In fact, if the bodies were unconstrained, their equations would have been written as:

$$\dot{p}^i_{Q_i} + S_1(w^i_{Q_i})p^i_{Q_i} = Q^i_{Q_i}$$

This fact suggests to define the multipliers λ^1_{Q1} and λ^2_{Q2}, which are related to the internal forces as momenta are to the external force, i e :

$$\lambda^{1}_{Q_1} - S_1(w^{1}_{Q_1})\lambda^{1}_{Q_1} = Q^{*}_{Q_1} \tag{8}$$

Substituting equation (8) into equation (6) it seems natural to define the modified momenta e.e.s as:

$$\tilde{p}^{1}_{Q_1} = p^{1}_{Q_1} - \lambda^{1}_{Q_1}$$

and to write equations (6) in the form:

$$\dot{\tilde{p}}^{1}_{Q_1} + S_1(w^{1}_{Q_1})\tilde{p}^{1}_{Q_1} - Q^{1}_{Q_1} = 0 \tag{9}$$

Equations (8) can be substituted into equation (7); thus establishing a linkage between the two multipliers. The result can be manipulated to obtain:

$$\frac{d}{dt}\left(\lambda^{1}_{Q1} + E\lambda^{2}_{Q2}\right) + S_1(w^{1}_{Q1})\left(\lambda^{1}_{Q1} + E\lambda^{2}_{Q2}\right) = 0 \tag{10}$$

Equation (10) is fulfilled by setting:

$$\lambda^{1}_{Q1} + E\lambda^{2}_{Q2} = 0 \tag{11}$$

The above equation suggests a transport law for the multipliers analogous to that used for the momenta. Moreover, starting from equation (11) and enforcing the definitions of the mutipliers, equation (7) can be obtained. This means that the set of equations (8) and (11) is fully equivalent to equation (7). Equations (9) can be referred to the reference points P_1 and P_2, the modified Euler equations can thus be obtained:

$$\dot{\tilde{p}}^{1}_{P_1} + S_1(w^{1}_{P_1})\tilde{p}^{1}_{P_1} - Q^{1}_{P_1} = 0 \tag{12}$$

The multipliers $\lambda^{1}_{P_1}$ and $\lambda^{2}_{P_2}$, referred to the points P_1 and P_2, can now be linked by applying the transport operators to equation (11); thus obtaining:

$$\lambda^{1}_{P_1} + H\lambda^{2}_{P_2} = 0 \tag{13}$$

The matrix H being the transport operator associated to the distance vector:

$$h = c_1 + e - c_2 = P_2 - P_1$$

Some interesting results have been obtained substituting the set of equations (6) and (7) with that of equations (8), (12) and (13). The equations of motion of each constrained rigid body can be written in the same way as those of the corresponding free body by exchanging the ordinary momenta with the modified ones; thus decoupling the equations of the constrained bodies. The above definitions of the modified momenta permit to project the Euler equations onto some directions of the phase space [22,23,24]. This projections have the property to free the modified momenta from any constraint. On the other hand the action of the constraint has been explicitly charged only onto equations (8) and (13) thus leaving equation (12) unrelated to any constraint.

A decoupling between the bodies can now be attempted also for equation (5) ruling the kinematics of the system through the constraint. To this end, let us refer the kinematic equation to the points P_1 and P_2.

$$w^2_{P2} - H^T w^1_{P1} - C_2^{-T} w^R_{Q2} = 0$$

the matrix C_2 being the transport operator associated to the distance vector c_2. A new unknown ν can be defined by:

$$w^1_{P1} - \nu = 0$$

and the kinematic equation through the constraint can be written as:

$$w^2_{P2} - H^T \nu - C_2^{-T} w^R_{Q2} = 0$$

To complete the system of DAE ruling the motion of the multibody system the constitutive equation of the constraint must be added. This equation has the form:

$$\mathcal{F}\left(Q^R_{Q2}, q^1_{P1}, w^1_{P1}, w^R_{Q2}, q^R_{P2}\right) = 0 \tag{14}$$

and cannot be further specified without entering into the details of each constraint.

4 Variational Principles

The aim of this paragraph is to develop a variational principle governing the motion of a multibody system. The starting point will be the system of DAE equations previously discussed and summarised in its most general form below, i.e.:

$$\dot{\tilde{p}}^1_{P_i} + S_1(\dot{w}^1_{P_i})\tilde{p}^1_{P_i} - Q^1_{P_i} = 0 \tag{15}$$

$$\lambda^1_{P1} + H\lambda^2_{P2} = 0 \tag{16}$$

$$\dot{w}^1_{P1} - \nu = 0 \tag{17}$$

$$\dot{w}^2_{P2} - H^T \nu - w^R_{P2} = 0 \tag{18}$$

$$\dot{\lambda}^2_{P2} + S_1(\dot{w}^2_{P2})\lambda^2_{P2} - Q^R_{P2} = 0 \tag{19}$$

$$\mathcal{F}\left(Q^R_{Q2}, w^1_{P1}, w^2_{P2}, w^R_{Q2}\right) = 0 \tag{20}$$

$$\dot{w}^1_{P_i} - \dot{w}^1_{P_i} = 0 \tag{21}$$

Equation (21) is the constitutive equation of the rigid bodies and will be considered only in the Hamiltonian form. Its role is to establish a linkage between the velocity w and $\dot{w}$ defined by:

$$w^1_{P_i} = X^{-1}\left(q^1_{P_i}\right)\dot{q}^1_{P_i} \qquad \dot{w}^1_{P_i} = M^{-1}_{P_i}\,\tilde{\rho}^1_{P_i}$$

in which $M^1_{P_i}$ represents the 6×6 mass matrix of the ith body. The variational principles will be achieved by multiplying the equations of the system by proper weight functions and combining them linearly. In a Lagrangian approach not all the equations and unknowns will be held. As a matter of fact, the constitutive equations (21) will be considered as satisfied a priori. The momenta e.v. will not be considered as a primary unknown but simply recovered a posteriori by multiplying the mass

matrix by the velocity. The velocity e.v. is related only to the time derivative of the Lagrangian coordinate e.v., so that, no difference exists between the velocity e.v. w and $\dot{w}$. Equation (15) can be scalarly multiplied by the pseudo-variation of the Lagrangian coordinates $q'_{P_1 i}$:

$$q'_{P_1 i}\left(\dot{\tilde{P}}'_{P_1} + S_1(w'_{P_1})\tilde{P}'_{P_1} - Q'_{P_1}\right) = 0$$

This equation can be manipulated by applying the product differentiation rule and writing $q'_{P_1 i}$ as depending from $\delta w'_{P_1}$; thus obtaining:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(q'_{P_1 i}\tilde{P}'_{P_1}\right) - \delta^{\oplus} w'_{P_1}\tilde{P}'_{P_1} - q'_{P_1 i}Q'_{P_1} = 0 \tag{22}$$

The Lagrangian function of a rigid body subjected to non conservative forces, can be written:

$$\mathcal{L}_1 = \frac{1}{2} w'_{P_1} M'_{P_1} w'_{P_1}$$

The second term of equation (22) suggests to define a modified Lagrangian function:

$$\tilde{\mathcal{L}}_1 = \mathcal{L}_1 - w'_{P_1}\lambda'_{P_1}$$

and to evaluate its variation:

$$\delta\tilde{\mathcal{L}}_1 = \delta^{\oplus}\tilde{\mathcal{L}}_1 = \delta^{\oplus} w'_{P_1}\tilde{P}'_{P_1} - w'_{P_1}\delta^{\oplus}\lambda'_{P_1}$$

This definitions can be introduced into equation (22) obtaining the variational form of the Euler equations.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(q'_{P_1 i}\tilde{P}'_{P_1}\right) - \delta\tilde{\mathcal{L}}_1 - w'_{P_1}\delta^{\oplus}\lambda'_{P_1} - q'_{P_1 i}Q'_{P_1} = 0 \tag{23}$$

The variational form for the differential system will be obtained by summing equations (23) with equations (16)-(20) scalarly multiplied by the functions $-\delta^{\oplus}\nu$, $\delta^{\oplus}\lambda^1_{P_1}$, $\delta^{\oplus}\lambda^2_{P_2}$ and $q^R_{P_2 i}$. Applying the product differentiation rule to all the terms containing an explicit time derivative of the unknowns and by manipulating the linear combination we can write:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\left(q^1_{P_1 i}\tilde{P}^1_{P_1} + q^2_{P_2 i}\tilde{P}^2_{P_2} + q^R_{P_2 i}\lambda^2_{P_2}\right) - \\
&-\delta\left(\tilde{\mathcal{L}}_1 + \tilde{\mathcal{L}}_2 + \nu\lambda^1_{P_1} + \left(H^T\nu + w^R_{P_2}\right)\lambda^2_{P_2}\right) + \\
&-\left(\sigma^1_{P_1 i}Q^1_{P_1} + q^2_{P_2 i}Q^2_{P_2} + q^R_{P_2 i}Q^R_{P_2}\right) + \\
&+ S_0^T(H^T\nu)\left(H^T q^1_{P_1 i} + q^R_{P_2 i} - q^2_{P_2 i}\right)\lambda^2_{P_2} = 0
\end{aligned} \tag{24}$$

It is interesting to note that, although from a mathematical point of view the weight functions must be independent in general sense, this independence is not completely arbitrary if a mechanical meaning is assigned to them. In fact, the last row of equation (24) must be zero if the kinematic consistency of the virtual displacements must be conserved. Some interesting remarks can be made on this variational form. At first it must be noticed that, substituting the ordinary Lagrangian functions and momenta e.v. with the modified ones, the contribution of each of the constrained rigid

bodies is the same as that of a free body: i.e., the sum of the boundary term of the variation of the Lagrangian function and of the virtual work of the external forces. Moreover, the constraint appears into the equation as a mechanical element characterised by a boundary and virtual work term for which the role of the Lagrangian coordinates is played by the relative coordinates: that of external forces by the internal forces and that of momenta by the constraint multipliers. The Lagrangian function cannot be defined for the constraint because it has no mass. The term corresponding to the variation of the Lagrangian function has been introduced in Routh's form: i.e., as the product of the generalised momenta e.v., i.e. the multipliers, with the correspondent generalised velocity e.v. [25,26] A similar principle can be obtained by operating on the Hamiltonian form of the system of DAE and it is not reported here for sake of brevity.

5 Linearized Weak Form

The weak form of the system of DAE is obtained by integrating over the time domain $[t_1, t_2] \subset R$ the dot product of the equations of motion in Hamiltonian form by some arbitrary test functions $v \in V$. Integrating by parts until all the time derivatives are wholly charged upon the test functions:

$$\int_{t_1}^{t_2} \left\{ \sum_{i=1}^{2} \left(-v_i' \dot{p}_{P_i}' + v_i' \left(S\left(\dot{w}_{P_i}'\right) \dot{p}_{P_i}' - Q_{P_i}' \right) + v_i' X^{-1}\left(q_{P_i}'\right) \dot{w}_{P_i}' + v_i' q_{P_i}' \right) + \right.$$

$$+ v_3 \left(\lambda_{P_1}^1 + H \lambda_{P_2}^2 \right) + v_4 \left(\dot{w}_{P_1}^1 - \nu \right) + \frac{\mathrm{d}^{\circledast}}{\mathrm{d}t} v_3 q_{Q_2}^{\wedge} +$$

$$+ v_3 \left(X^{-1}\left(q_{Q_2}^{\wedge}\right) C_2^T \dot{w}_{P_2}^2 - X^{-1}\left(q_{Q_2}^{\wedge}\right) C_2^T H^T \nu \right) +$$

$$\left. - \dot{v}_4 \lambda_{P_2}^1 + v_4 \left(S_1(\dot{w}_{P_2}^2) \lambda_{P_2}^2 - Q_{P_2}^{\wedge} \right) + v_7 \mathcal{F}\left(Q_{Q_2}^{\wedge}, w_{P_1}^1, w_{P_2}^2, w_{Q_2}^{\wedge} \right) \right\} dt =$$

$$\left[\sum_{i=1}^{2} \left(-v_i' \dot{p}_{P_i}' + v_i' q_{P_i}' \right) + v_3 q_{Q_2}^{\wedge} - v_4 \lambda_{P_2}^1 \right]_{t_1}^{t_2} \tag{25}$$

To introduce the numerical approximation, we write the weak form (25) in a more abstract way. To this end, let us call $\mathcal{V}$ the vector formed by all the test functions and by their time derivatives, $\mathcal{J}(u)$ the vector gathering all the functions multiplied by them and $\mathcal{B}$ the scalar function containing all the terms at the boundary of the time domain. Equation (25) can thus be written:

$$\int_{t_1}^{t_2} \mathcal{V} \mathcal{J}(u)\, \mathrm{d}t = \{\mathcal{B}\}_{t_1}^{t_2} \tag{26}$$

The test functions appear into equation (26) with their time derivatives and the trial functions appear through the vector $\mathcal{J}(u)$ without any time derivative. A possible choice of the space of the test functions could be:

$$V := \left\{ v(t) : v \in H^1(t_1, t_2) \right\} \qquad U := \left\{ u(t) : u \in H^0(t_1, t_2) \right\}$$

with appropriate boundary conditions for the unknowns which are defined on the boundary. $V \subseteq U$ and this fact will permit, as shown in the next paragraph, the use of two different numerical approximations for the test and trial functions.

The weak form (26) is linearly dependent upon the test functions and their derivatives and it is non linearly dependent upon the unknown vector u. In order to apply the finite elements in time, the

weak form (26) must be written as a bilinear form in the test and trial functions. To this end, we linearize the vector $\mathcal{J}(u)$ around a reference condition $\overline{u}$. It is noted that $\mathcal{B}$ does not need to be linearized

$$\int_{t_1}^{t_2} \mathcal{V}\mathcal{J}(u)\,dt = \int_{t_1}^{t_2} \mathcal{V}\mathcal{J}(\overline{u})dt + \int_{t_1}^{t_2} \mathcal{V}\mathcal{T}(\overline{u})\,\mathrm{d}u dt + \int_{t_1}^{t_2} \mathcal{V}\varepsilon dt \tag{27}$$

$\mathcal{T}(\overline{u})$ is the jacobian matrix of the vector $\mathcal{J}(u)$ evaluated at $\overline{u}$, $\mathrm{d}u$ is the increment of the unknowns and ε all the terms that will be neglected. The incremental equation can be obtained by substituting equation (27) into (26) and omitting all the negligible terms.

$$\int_{t_1}^{t_2} \mathcal{V}\mathcal{T}(\overline{u})\,\mathrm{d}u dt = [\mathcal{B}]_{t_1}^{t_2} - \int_{t_1}^{t_2} \mathcal{V}\mathcal{J}(\overline{u})\,dt \tag{28}$$

6 Numerical Approximation

The aim of this paragraph is to introduce a numerical discretization and to describe the solution algorithm for a typical initial value problem. The vector $\mathcal{V}$ in equation (28) is ordered with the test functions followed by their time derivatives. The matrices $\mathcal{T}(\overline{u})$ and $\mathcal{J}(\overline{u})$ are then ordered consequently. Thus, indicating with $\mathcal{T}_1(\overline{u})$ and $\mathcal{J}_1(\overline{u})$ the blocks of $\mathcal{T}(\overline{u})$ and $\mathcal{J}(\overline{u})$ multiplied by the test functions and with $\mathcal{T}_2(\overline{u})$ and $\mathcal{J}_2(\overline{u})$ those multiplied by their time derivatives, equation (28) becomes:

$$\int_{t_1}^{t_2} \begin{Bmatrix} v \\ \dot{v} \end{Bmatrix} \begin{bmatrix} \mathcal{T}_1(\overline{u}) \\ \mathcal{T}_2(\overline{u}) \end{bmatrix} \mathrm{d}u dt = [v_s u_s]_{t_1}^{t_2} - \int_{t_1}^{t_2} \begin{Bmatrix} v \\ \dot{v} \end{Bmatrix} \begin{bmatrix} \mathcal{J}_1(\overline{u}) \\ \mathcal{J}_2(\overline{u}) \end{bmatrix} dt \tag{29}$$

In this equation we apply the finite elements approximation after dividing the time domain $[t_1, t_2]$ in n time finite elements. The test and the trial functions v and u are then written as.

$$v = \phi_V(t)V \qquad u = \phi_U(t)U$$

in which V and U represent the vectors of the test and unknown functions at the nodes. ϕ_V and ϕ_U are basis in the spaces $U_h \subseteq U$ and $V_h \subseteq V$ in which the approximated problem is solved. If a polynomial approximation is used, the spaces U_h and V_h are subspaces of the space $\mathcal{S}_h$ of the kth order polynomials. As shown in the last paragraph, the functions ϕ_V can be 1st order polynomials at least, i.e. piecewise linear functions. ϕ_U can instead be 0 order polynomials at least, i.e. piecewise constants. The number of nodes N_V defining the test functions (at least two) and those N_U used for the unknowns (at least one) can thus be related by:

$$N_V = N_U + 1$$

Substituting such an approximation into equation (29), it becomes:

$$V\left(\int_{t_1}^{t_2} \begin{Bmatrix} \phi_V \\ \dot{\phi}_V \end{Bmatrix} \begin{bmatrix} \mathcal{T}_1(\overline{u}) \\ \mathcal{T}_2(\overline{u}) \end{bmatrix} \phi_U dt\right) \mathrm{d}U = [v_s u_s]_{t_1}^{t_2} - V\left(\int_{t_1}^{t_2} \begin{Bmatrix} \phi_V \\ \dot{\phi}_V \end{Bmatrix} \begin{bmatrix} \mathcal{J}_1(\overline{u}) \\ \mathcal{J}_2(\overline{u}) \end{bmatrix} dt\right) \tag{30}$$

On the other hand, the shape functions that multiply the terms u_s are non zero only at the two external nodes. The boundary terms can thus be written:

$$[v, u_s]_{t_1}^{t_2} = V \begin{bmatrix} -I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} u_{s1} \\ u_{s2} \end{Bmatrix} \tag{31}$$

Indicating with n_v the number of unknowns and with n_{t_s} the unknowns defined at the boundaries. I a unit diagonal matrix with n_{t_s} rows. the central block is formed by $n_v \cdot (n-1)$ null rows. Substituting equation (31) into (30):

$$\left(\int_{t_1}^{t_2} \begin{Bmatrix} \phi_v \\ \dot{\phi}_v \end{Bmatrix} \begin{bmatrix} \mathcal{T}_1(\overline{u}) \\ \mathcal{T}_2(\overline{u}) \end{bmatrix} \phi_v dt \right) \mathrm{d}U = \begin{bmatrix} -I & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} u_{s1} \\ u_{s2} \end{Bmatrix} - \int_{t_1}^{t_2} \begin{Bmatrix} \phi_v \\ \dot{\phi}_v \end{Bmatrix} \begin{bmatrix} \mathcal{J}_1(\overline{u}) \\ \mathcal{J}_2(\overline{u}) \end{bmatrix} dt \tag{32}$$

For a typical initial value problem equation (32) represents a system of linear equations in the unknowns $\mathrm{d}U$ and u_{s2} with known terms defined by the first boundary condition and by the tentative solution $\overline{u}$. The linear system can thus be written in the form:

$$\begin{bmatrix} \mathcal{K} & 0 \\ \mathcal{K}_s & -I \end{bmatrix} \begin{Bmatrix} \mathrm{d}u \\ u_{s2} \end{Bmatrix} = \begin{Bmatrix} \mathcal{R} \\ \mathcal{R}_s \end{Bmatrix} + \begin{bmatrix} -I \\ 0 \\ 0 \end{bmatrix} u_{s1} \tag{33}$$

with:

$$\begin{bmatrix} \mathcal{K} \\ \mathcal{K}_s \end{bmatrix} = \int_{t_1}^{t_2} \begin{Bmatrix} \phi_v \\ \dot{\phi}_v \end{Bmatrix} \begin{bmatrix} \mathcal{T}_1(\overline{u}) \\ \mathcal{T}_2(\overline{u}) \end{bmatrix} \phi_v dt \qquad \begin{Bmatrix} \mathcal{R} \\ \mathcal{R}_s \end{Bmatrix} = - \int_{t_1}^{t_2} \begin{Bmatrix} \phi_v \\ \dot{\phi}_v \end{Bmatrix} \begin{bmatrix} \mathcal{J}_1(\overline{u}) \\ \mathcal{J}_2(\overline{u}) \end{bmatrix} dt$$

We observe that the upper block of equations in (33) is not affected by the unknowns u_{s2} but only by $\mathrm{d}U$. This fact suggests an iterative solution algorithm. Starting from a trial solution U_0, the increment $\mathrm{d}U$ can be evaluated by solving the upper side of the linear system (33). The solution is updated at any iteration:

$$U_{i+1} = U_i + \mathrm{d}U$$

U_{i+1} becomes the new reference solution to evaluate the tangent matrix $\mathcal{K}$ and the residual vector $\mathcal{R}$. Once the convergence is reached, the increment $\mathrm{d}U$ is considered negligible. In such a way, it cannot affect the lower side of system (33) and the final boundary condition can be directly recovered by:

$$u_{s2} = -\mathcal{R}_s(U_{i+1}) \tag{34}$$

This procedure can be applied in the assembled form on the time domain $[t_1, t_2]$ or in the incremental form on the single time finite element $[t_i, t_{i+1}]$. In this case the final boundary condition becomes the initial one for the successive time step. It must also be observed that not all of the unknowns have boundary terms, in fact, the vectors λ_{p1}^1, ν and Q_{Q2}^R have only internal values. On the other hand, not all the equations are integrated by part so that the balance between equations and unknowns is conserved. Considering a general multibody system formed by N_B bodies and N_C constraints. the total number of internal and boundary unknowns is the given by:

$$N = 6 \cdot (2 \cdot N_V N_B + 2 \cdot N_V N_C + 3 \cdot N_U N_C) \tag{35}$$

It should be remarked that the subsequent examples have been carried out by assuming $N_U = 1$ and $N_V = 2$: this should be understood, even if it is not explicitly stated.

7 Sparse Matrix Techniques

Equation (35) makes it clear that the advantages of a more than redundant formulation implies a heavy cost in term of the number of unknowns. This fact does not implies a correspondent increase in computer cost because of the high sparsity of the resulting nonlinear system of equations. In fact, most of the equations are nothing more than mere definitions adopted to allow the decoupling of the equations and are thus related just to few unknowns. Moreover their elimination during the solution, e.g. through a factorization, requires very few additions and multiplications and maintains the sparsity of the system since it introduces new terms whose number is of the order of the directly connected bodies. It should be sufficient to note that the multibody system presented in Example 2.2 requires 1032 unknowns, 372 of which are recovered a posteriori as boundary terms by means of equation (34). The remaining 660 unknowns present a sparsity of .81 per cent that, after the factorization, becomes of 1.15 per cent. It is noted that the approach presented in this paper has many similarities with the tableau techniques adopted in electrical circuits analysis [27]. Thus, despite the large number of unknowns, the more than redundant coordinates formulation can be treated efficiently by the adoption of sparse solvers of the same type used for circuits analysis [28]. In the implementation of the present formulation the solver has been taken from Reference 29 and the assembling of the sparse system has been programmed anew through an efficient implementation of the hashing techniques found in the same reference. Hashing is quite effective since, because of the spare memory that must conservatively be kept in view of the fill in during the factorization, the working core space can be allocated once for all and the probability of different matrix terms colliding is very rare.

It can be interesting to note that, since the assembly and solution time is negligible with respect to that of setting up the equations, it is easy to implement the present formulation on parallel computers. In such a case the equations of the different bodies and constraints can be evaluated independently on the available CPUs without any need of synchronisation. The assembly can also be easily paralleled by pre assembling the elements to be processed by each CPU. The only operations to be performed in series are the final assemblies of the sub blocks and the solutions which, as previously noted, are of negligible cost. It is furthermore noted that what said above is applicable both in case of shared and dedicated central memory as communication and/or synchronisation of the different CPU's is required only during the final phase.

8 Numerical Examples

In order to prove the capability of the present method in handling complex problems in rigid multibody dynamics, some examples have been considered. The results will be discussed and compared to those available in the literature as well as to those obtained by using the multibody program *ADAMS* [30].

8.1 Example 1: The Bipendulum

We have considered the motion of the bipendulum of Figure 2 whose geometrical and inertial properties are reported in Table 1. At time $t = 0$ the two bars are aligned on the negative y axis and rotate with an anguiar velocity of 4 rad/s around the z axis. Two different configurations have been tested considering different combinations of joints.

In Example 1.1 the bars are linked by spherical joints and the result is a very complex three dimensional motion because of the combination of the weight, centrifugal and gyroscopic forces. Figure 3

presents the cross plot of the coordinates x, y of the centre of mass (CoM) of the two bars and allows to appreciate the perfect accordance between these results and those taken from the literature [21,31]
In Example 1.2 bar 1 is connected to the IRF by means of a constraint that enforces the component of the relative angular velocity along the z axis to be constant. The linkage between bars 1 and 2, instead, is constituted by a damped hinge with an hinge axis parallel to the x direction at time $t = 0$ and embedded into body 1. The motion of bar 2 is a damped oscillation into the plane yz till a position is reached in which the inertia forces balance the weight. Such a position can be evaluated by solving a problem with a single degree of freedom. Figure 4 presents the time history of the z coordinate of the CoM of bar 2 for some values of the damping coefficients c. Despite the lack of comparisons, the configuration of dynamic equilibrium and the behaviour of the system at the changing of the damping coefficient, have been verified to be correct.

8.2 The Fifteen Links Chain

These examples have been taken from a classical test case of multibody dynamics, i.e. the fifteen links chain of Figure 5. Each link of the chain is a bar as in the previous example and the joints between the links have been modelled by means of spherical joints. In the initial configuration the chain is suspended at points 1, 6, 11 and 16 and it is then left to fall unhooking some of the suspension points.
In Example 2.1 the chain falls suspended only at point 1 for 1.5 seconds. Figure 6 presents the animation of the fall while Figure 7 presents the time history of the z coordinate of the CoM of bars 3, 8, 13. Figure 8 presents the time history of the norm of the internal force in joint 8. The results are in accordance with both those in the reported literature [23,31] and with those given by $ADAMS$. The action of the constraints is progressively extended from the left to the right of the chain. Each link falls substantially free, the x coordinates describe the same parabola, till the left side neighbouring recalls it in an almost impulsive way as shown by the peaks in the constraint reaction.
In Example 2.2 the chain falls suspended at points 1 and 16 for 0.5 seconds and the bars 1 and 15 are enforced to rotate counter clockwise around the x axis with a constant angular velocity of 20 rad/s. The problem, without any comparison in the literature, constitutes a tough test for the robustness of the method. Figures 9a presents the animation of the chain in the time interval from 0 to .16 seconds. The links of the side arches rotate and press on the mid arch, thus slackening the falls of its links, as shown by Figure 10. Figures 9b presents the animation during the time interval from .16 to .33 seconds, in which the elements of the middle arch are sharply flung up. The time history of the CoM of bar 8 presents an angular point and a strong increase of the slope. Figures 9c presents the motion in the time interval from .33 to .5 seconds. The elements of the side arches are folded and rotate around the points 1 and 16; thus tightening the chain. It must be emphasised that the examples have been carried out as three dimensional simulations, although Example 1.2 presents an axial symmetry and examples 2 present planar motion. This fact has permitted to verify the robustness of the algorithm in conserving some geometric properties as planarity or symmetry. The results are in accordance with those of the $ADAMS$ simulation during the first phase. The angular point on the bottom of the trajectory is slightly shifted but the flung up presents the same slope No folding is observed for the elements of the external arches and thus, those of the central, present a wide oscillation without to be tightened. It must be observed that discrepancies emerged only for symmetric problems, in which the motion is so constricted to present significant bifurcations in the unknowns space. In these situations the behaviour of the system is difficult to be predicted because of the non linearities and appears to be intimately related to the numerical approximation and to the solution algorithm.

9 Concluding Remarks

In this work presented a general algorithm for the dynamic analysis of rigid multibody systems. The theoretical analysis of the kinematic and dynamic models permitted to set the problem in an effective working frame. Starting from the system of differential-algebraic equations, a variational principle was developed that allowed an interpretation of the present work from the classical point of view of analytical mechanics. A proper definition of the multipliers allowed to achieve the double aim of an intrinsic stabilisation of the constraints, even in presence of numerically stiff problems, and a total decoupling of the equations of the bodies and the constraints. This decoupling can be profitably and easily exploited if parallel computation techniques can be applied to increase the efficiency of the method. The use of relative coordinates and of internal forces makes it natural the imposition of many and different constraints: thus permitting an easy development of a wide constraints library as requested by multibody dynamics. On the other hand, despite the large number of unknowns and equations, an efficient solution has been achieved by the adoption of a sparse solver. Moreover, the present work provides a sound basis for a further development of the method. In this view, the addition of deformable elements and active control techniques seem to be the most significant.

References

[1] Klimov D.M. Banichuk N.V. and Schielen W., editors. *Dynamical Problems of Rigid-Elastic Systems and Structures*. IUTAM Symposium. Springer-Verlag, Moscow. USSR. May 23-27 1991

[2] Schielen W.O., editor. *Multibody Systems Handbook*. Springer-Verlag, 1990.

[3] E.J.Haug and R.C.Deyo, editors. *Real-Time Integration Methods for Mechanical System Simulation*. NATO Advanced Research Workshop. Springer-Verlag, Snowbird. USA. Aug 7-11 1989. NATO ASI Series F: Computer and System Sciences, vol.69.

[4] M.Geradin and O.Friberg, editors. *Computer Methods in Flexible Multibody Dynamics*. International Journal for Numerical Methods in Engineering, 1991. Special Issue enclosed in vol.32.

[5] Roberson R.E. and Schwertassek R. *Dynamics of Multibody Systems*. Springer-Verlag, Berlin, 1988.

[6] Kane T.R. and Wang C.F. On the Derivation of Equatiions of Motion. *J. Soc. Indust. Appl. Math.*, 13(2):487–492, June 1965.

[7] Huston R.L. and Passarello C.E. Multibody Structural Dynamics Including Transiation Between the Bodies. *Computers & Structures*, 12:713–720, 1980.

[8] Nikravesh P.E. Systematic Reduction of Multibody Equations of Motion to a Minimal Set. *Int. J. Non-Linear Mechanics*, 25(2/3):143–151, 1990.

[9] Boland P., Samin J.C., and Willems P.Y. Stability Analysis of Interconnected Deformable Bodies with Closed-Loop Configuration. *AIAA Journal*, 13(7):864–867, 1975.

[10] Orlandea N., Chase M.A., and Colaham D.A. A Sparsity Oriented Approach to the Dynamic Analysis and Design of Mechanical System, Parts I and II. *ASME Journal of Engineering for Industry*, 99:773–784, 1977.

[11] Brauchli M.A. and Weber R.W. Canonical Approach to Multibody Systems Using Redundant Coordinates. In *Dynamics of Multibody Systems*, pages 31–41, IUTAM/IFoMM Symposium. 1985.

[12] Rampalli R ADAMS-A Sparse Matrix Approach to Solving Multibody Dynamic Problems. In *SDIO/NASA Workshop on Multibody Simulation*, Jet Propulsion Laboratory, Pasadena, Cal., 1987.

[13] Singh R.P. and Likins P.W. Singular Value Decomposition for Constrained Dynamical Systems. *Journal of Applied Mechanics*, 52.943–948, 1985.

[14] Mani N.K. Application of Singular Value Decomposition for Analysis of Mechanical System Dynamics. *American Society for Mechanical Engineers*, 84-DET-89:1–6, 1989.

[15] Amirouche F.M. and Jia T. Automatic Elimination of the Undetermined Multipliers in Kane's Equations Using a Pseudo–Uptriangular Decomposition. *Computers & Structures*, 27(2).203–210, 1987.

[16] Kim S.S. and Vanderploeg M.J. QR Decomposition for State-Space Representation of Constrained Mechanical Dynamic Systems. *Journal of Mechanisms, Transmission and Automation in Design*. 108:183–188, 1986.

[17] Ramesh A.V., Utku S., and Wada B.K. Parallel Computation of Geometry Control in Adaptative Truss Structures. *Journal of Intelligent Materials Systems and Structures*, 3:483–500, July 1992.

[18] Gear W. Parallel Solution of ODEs. In E.J.Haug and R.C.Deyo, editors, *Real-Time Integration Methods for Mechanical System Simulation*. NATO Advanced Research Workshop, Springer-Verlag, Snowbird, USA, Aug 7-11 1989. NATO ASI Series F: Computer and System Sciences, vol.69.

[19] Curtis M.L. *Matrix Groups*. Springer-Verlag, Berlin, 1984. 2nd edition.

[20] Hughes P.C. *Spacecraft Attitude Dynamics*. John Wiley and Sons, New York, 1986.

[21] Mello F. *Weak Formulations in Analytical Dynamics, with Applications to Multi-Rigid-Body Systems, Using Time Finite Elements*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, December 1989.

[22] Borri M., Bottasso C., and Mantegazza P. Acceleration Projection Method in Multibody Dnamics. *European Journal of Mechanics*, 11(3):403–418, 1992

[23] Borri M., Bottasso C., and Mantegazza P. A Modified Phase Space Formulation for Constrained Mechanical Systems – Differential Approach. *European Journal of Mechanics*, 11(5):701–727, 1992.

[24] Blajer W. A Projection Method Approach to Constrained Dynamic Analysis. *Journal of Applied Mechanics*, 59:643–649, September 1992.

[25] Levi-Civita T. and Amaldi U. *Lezioni di Meccanica Razionale*. Zanichelli Editore, Bologna, 1926. In Italian.

[26] Pars L.A. *A Treatise on Analytical Dynamics*. OX Bow Press, Woodbridge, Connecticut, 1979

[27] Chua L.O., Desoer C.H., and Kuh E.S. *Linear and Nonlinear Circuits*. McGraw-Hill, 1987.

[28] Kundert K.S. Sparse Matrix Techniques and their Application to Circuit Simulation. In Ruehli A.E., editor, *Circuit Analysis, Simulation and Design*, pages 281–324, North Holland, 1988

[29] *Harwell Subroutine Library*. Advanced Computing Library, Harwell Laboratory, Oxfordshire, England, 2nd edition, September 1990.

[30] *ADAMS Users Guide*. Mechanical Dynamics Inc., 3055 Plymouth Rd., Ann Arbor, Michigan, 1987.

[31] Franchi C.G. A More than Redundant Coordinates Formulation for Constrained Rigid Bodies. *submitted for publication on Meccanica*, 1992.

[32] Franchi C.G. A General Formulation in Rigid Multibody Dynamics. *submitted for publication on Int. J. for Num. Meth. in Eng.*, 1993.

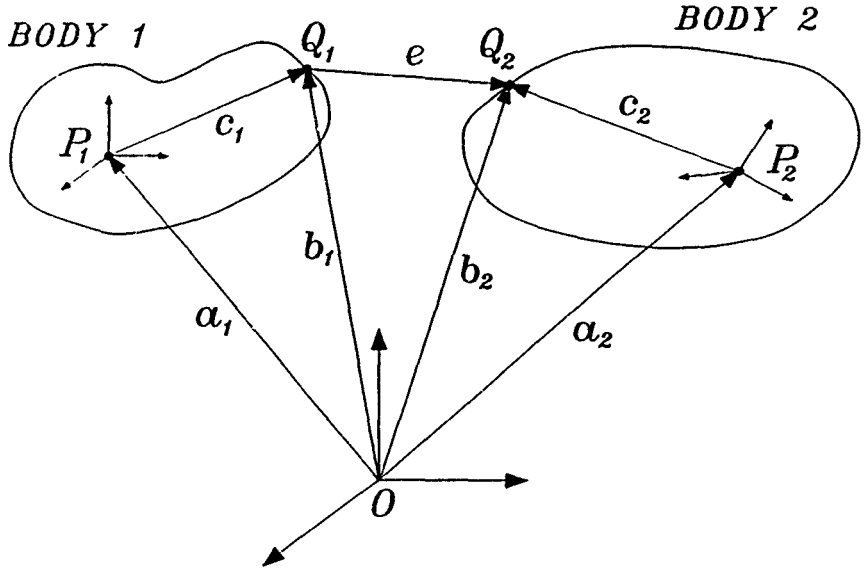


FIGURE 1

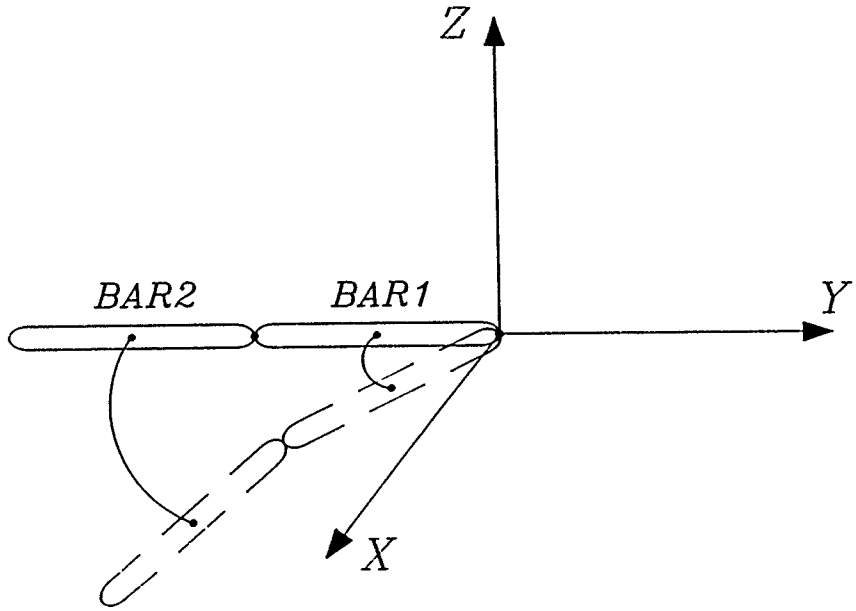


FIGURE 2

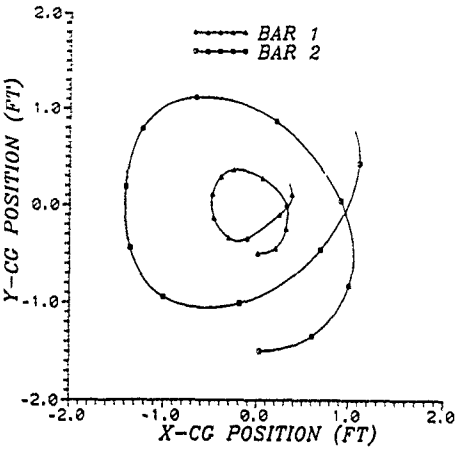


FIGURE 3

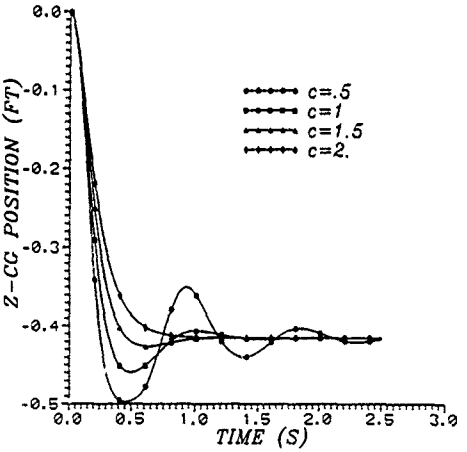


FIGURE 4

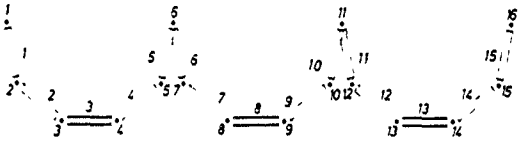


FIGURE 5

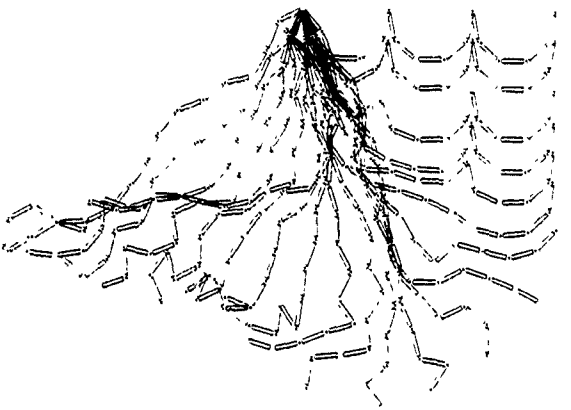


FIGURE 6

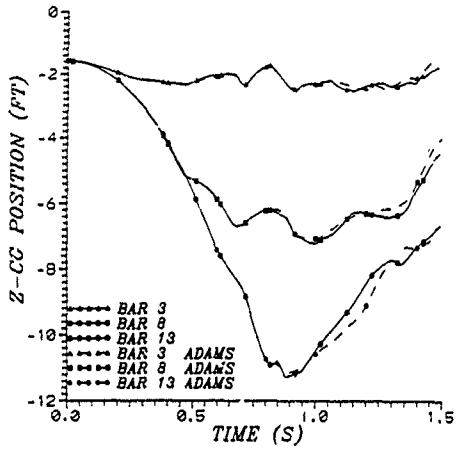


FIGURE 7

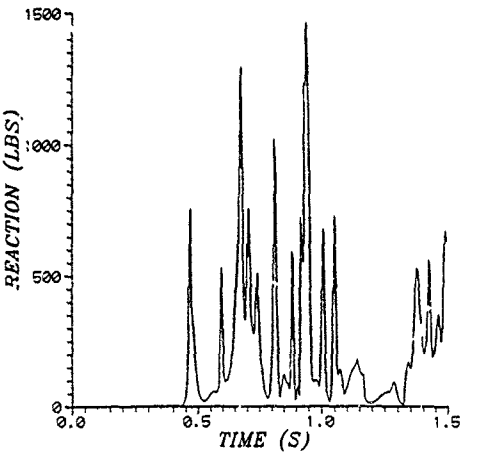


FIGURE 8

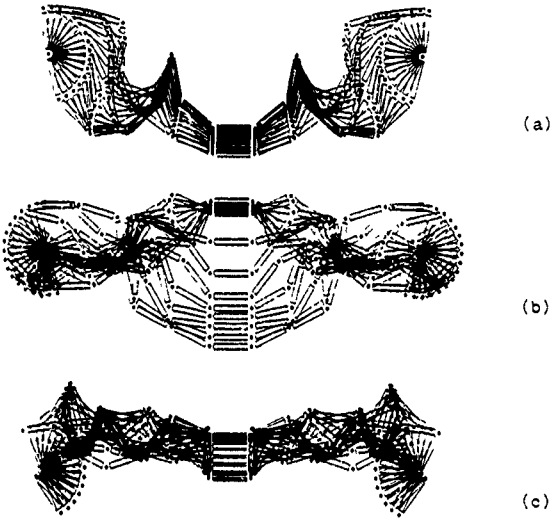


FIGURE 9

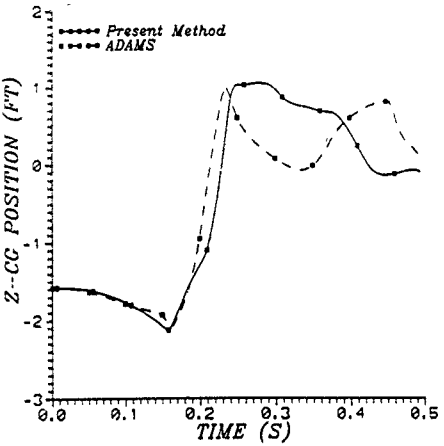
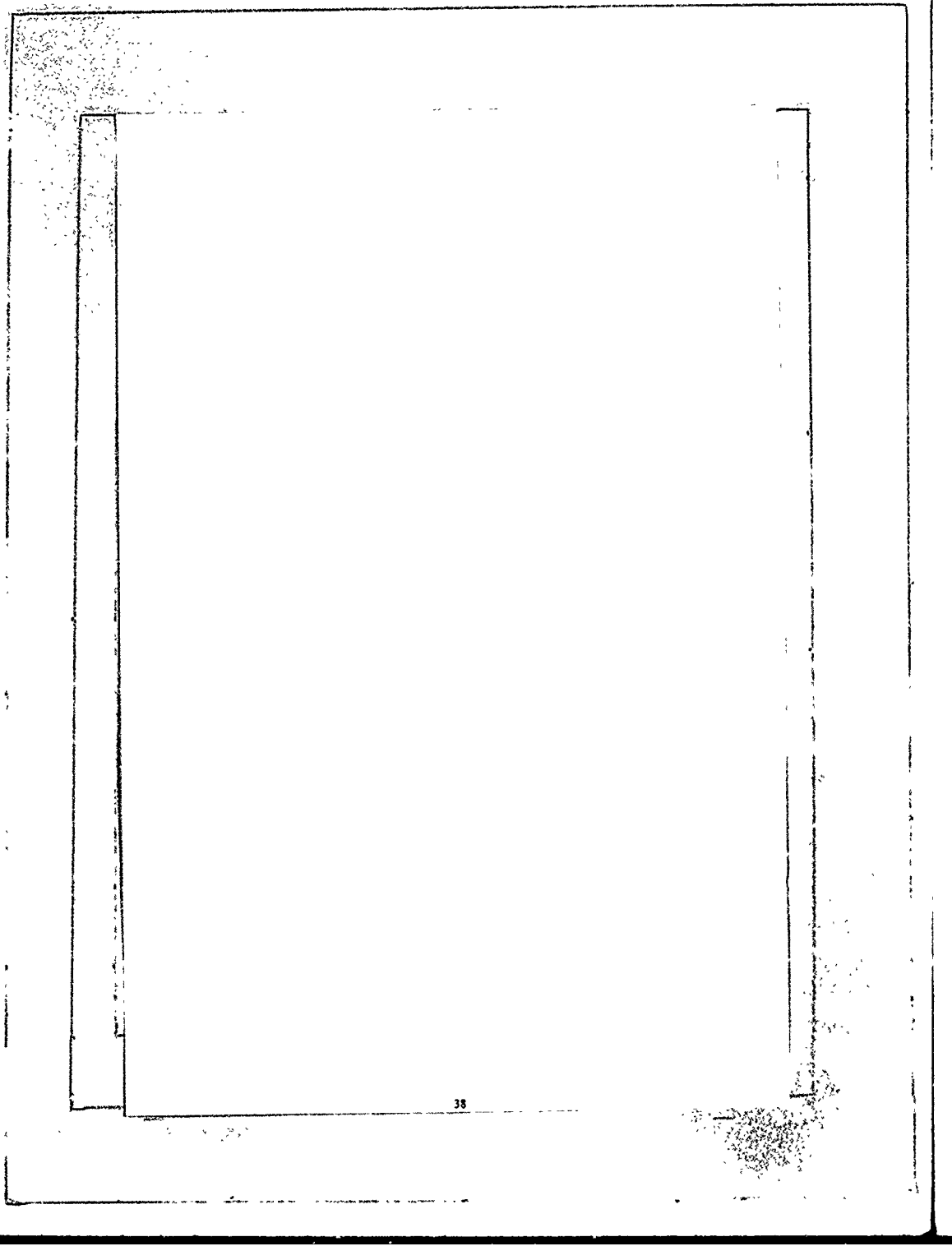

Present Method
ADAMS
Z--CG POSITION (FT)
2
1
0
-1
-2
-3
0.0
0.1
0.2
0.3
0.4
0.5
TIME (S)

FIGURE 10

Mass	.4565
$J_{trans.}$	.0308
J_{axial}	.0001
gravity	32.174
length	1.000

Table 1: Bipendulum Data



AN EFFICIENT IMPLEMENTATION OF THE VELOCITY TRANSFORMATION METHOD FOR REAL-TIME DYNAMICS WITH ILLUSTRATIVE EXAMPLES

J. M. JIMÉNEZ, A. N. AVELLO, J. GARCÍA DE JALÓN AND A. L. AVELLO
University of Navarra and CEIT
P. Manuel de Lardizábal 15
20009 San Sebastián
Spain

ABSTRACT. This paper presents an efficient algorithm based on velocity transformations for real-time dynamic simulation of multibody systems. Closed-loop systems are turned into open-loop systems by cutting joints. The closure conditions of the cut joints are imposed by explicit constraint equations. An algorithm for real-time simulation is presented that is well suited for parallel processing. The most computationally demanding tasks are matrix and vector products that may be computed in parallel for each body. Four examples are presented that illustrate the performance of the method.

1. Introduction

Real-time simulation is becoming steadily more important. With the appearance of the first digital computers the challenge of simulating multibody systems started. The first challenge was to develop general and robust formulations well suited for computer implementation. Several multibody software programs were developed as a result of these investigations. Most of them are described in Schiehlen (1990). From the sixties to the late eighties, the programs were steadily improved by adding new capabilities. Nowadays, the multibody simulation programs are powerful and versatile tools, able to carry out simulations of systems with all kinds of joints and subject to any type of applied forces. Furthermore, they include interfaces with CAD, finite element and animation programs. However, many of these programs have sacrificed efficiency on behalf of generality and versatility.

Real-time multibody simulation is the starting point for man-in-the-loop simulators and hardware-in-the-loop testing and design set-ups. To achieve real-time behavior it is necessary to have a sufficiently fast computer but also an efficient numerical algorithm. Processor speeds are increasing by a factor of 3 in only a few years and at the same time processors' architectures are evolving incessantly. However, formulating and solving the dynamics of multibody systems is an inherently complex task that requires very efficiently and well-implemented algorithms.

A thorough survey of dynamic formulations for serial rigid body multibody systems is given by Jain (1991). Formulations whose complexity grows linearly with the number of degrees of freedom are called $O(N)$ formulations. Jain (1991) reviews most of the $O(N)$, $O(N^2)$ and $O(N^3)$ formulations and gives a unified approach for serial open-loop systems. Well-known formulations, such as those developed by Walker and Orin (1982) Featherstone (1987),

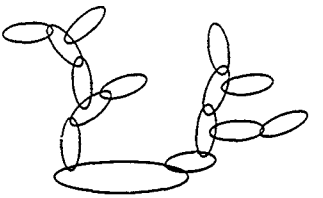


Figure 1. Open chain multibody system.

Armstrong (1979), Bae and Haug (1987) and Rosenthal (1987), are among the formulations reviewed.

Fijany and Beczy (1990) presented an improved version of Walker and Orin's (1982) $O(N^2)$ method and developed an algorithm well suited for parallel implementation on massively parallel computers.

Bae and Haug (1987) presented a generalization of Featherstone's $O(N)$ method for systems with all types of joints. Bae et al (1988) showed the potential of this method for parallel processing. Chang and Kim (1989) used this method to build a backhoe simulator based on a workstation.

Kim and Vanderploeg (1986) used the concept of *velocity transformations*, used first by Jerkovsky (1978), to develop an efficient $O(N^3)$ method. A similar method was also used by Huston (1990). García de Jalón et al. (1989) and Avello et al (1993) showed the potential of Kim and Vanderploeg's formulation for parallel processing on shared memory workstations. This paper uses the main ideas presented in Avello et al. (1993), with slight modifications, and presents a series of illustrative examples with open and closed-loops. The efficiency of the formulation is tested by comparing the CPU times of this formulation with the formulation used in the multibody program COMPAMM.

2. Kinematic Analysis

In this section we present efficient recursive solutions for the position, velocity and acceleration problems of multibody systems composed of an arbitrary number of joints and bodies. In addition to a global (or inertial) reference frame, consider a moving reference frame rigidly attached to each body. The position of the multibody system is characterized by a vector **x**, composed of

- Coordinates **r** of some points located at the joints
- Components **u** of unit vectors pointing in the direction of the joint axes
- Coordinates **g** of the centers of gravity
- 3x3 rotation matrices **A** relating the global frame with the moving frame of each body

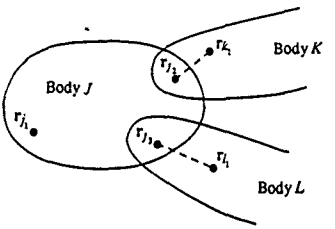


Figure 2. Generic body in a tree-structure system

Let z be the vector of F independent joint coordinates and base body variables, where F is the number of degrees of freedom. For the sake of efficiency we will assume that all the body reference frames are initially parallel. If they are not, the formulation presented in this section can easily be generalized by simply applying constant rotation matrices to the transformations of all the body vectors.

2.1. KINEMATICS OF OPEN-CHAIN MULTIBODY SYSTEMS

Consider a multibody system with tree structure, similar to the one presented in figure 1. The multibody is composed of a base body and a series of branches that can have arbitrary size and distribution and are assumed to be composed of rigid bodies interconnected by kinematic joints.

The paths going from the base body to the distal bodies are called forward paths. A certain body can be reached from only one body along a forward path, except the base body. One or more forward paths leave from each body except from the distal ones. In other words, all the bodies have a father, except the base body, and all the bodies have one or more children except the distal bodies.

The position problem consists in finding the dependent coordinates x given the independent coordinates z. This can be done in a recursive manner, starting from the base body and moving forward towards the distal elements in the different branches.

As mentioned earlier, x is composed of the coordinates of some points and vectors located at the joints, as well as of the coordinates of the centers of gravity and the rotation matrices between the body reference frames and the inertial frame.

Figure 2 shows a generic body J that has an input point $\mathbf{r}_{j_1}$ located at the joint between itself and its father and one or more output points $\mathbf{r}_{j_2}$, $\mathbf{r}_{j_3}$, etc., located at the joints between itself and its children.

Figure 3 illustrates a generic joint between body J and one of its children, body K. The body points $\mathbf{r}_{j_2}$ and $\mathbf{r}_{k_1}$, corresponding to bodies J and K, respectively, are located at the joint between the two bodies. Similarly, the body vectors $\mathbf{u}_{j_2}$ and $\mathbf{u}_{k_1}$ belong to J and K, respectively, and point in the direction of the joint axis. Some joints, for example the spherical joint, do not have an axis and therefore the two vectors are not necessary. For the moment we

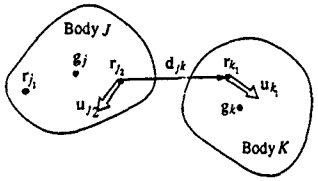


Figure 3. Generic joint model between bodies J and K.

will maintain the two vectors and later on we will particularize and simplify the expressions for the different types of joints.

The joint's offset is given by $\mathbf{d}_{jk}$, the vector from r_{j_2} to r_{k_1}. The joint's rotation is represented by the 3x3 rotation matrix $\mathbf{A}_{jk}$. Both $\mathbf{d}_{jk}$ and $\mathbf{A}_{jk}$ depend on the joint variables $\mathbf{z}_{jk}$ Assuming that the position of body J and the joint variables $\mathbf{z}_{jk}$ are known, we can compute the rotation matrix and the position of all the points and vectors of body K. The expressions to obtain them are

$$\mathbf{r}_{k_1} = \mathbf{r}_{j_1} + \mathbf{A}_j(\bar{\mathbf{r}}_{j_2} - \bar{\mathbf{r}}_{j_1}) + \mathbf{d}_{jk} \tag{1}$$

$$\mathbf{A}_k = \mathbf{A}_j \mathbf{A}_{jk} \tag{2}$$

$$\mathbf{g}_k = \mathbf{r}_{k_1} + \mathbf{A}_k(\bar{\mathbf{g}}_{k_1} - \bar{\mathbf{r}}_{k_1}) \tag{3}$$

$$\mathbf{r}_{k_2} = \mathbf{g}_k + \mathbf{A}_k\left(\bar{\mathbf{r}}_{k_2} - \bar{\mathbf{g}}_k\right) \tag{4}$$

$$\mathbf{u}_{k_1} = \mathbf{A}_k \bar{\mathbf{u}}_{k_1} \tag{5}$$

where the upper bar denotes coordinates expressed in the body reference frame. Note that equations (4) and (5) also apply to any other point or vector of body J. Equations (1) through (5) can be used recursively, following forward paths, to solve the position problem.

It is possible, and even convenient, to use a set of dependent velocities and accelerations that does not correspond to the derivatives of the position vector x. We will take as dependent velocities the velocity of the center of gravity and the angular velocity of each body. Consider again the generic joint of figure 3. Assuming that the linear and angular velocity of the body J is known, we can compute the velocity of K as follows

$$\dot{\mathbf{r}}_{k_1} = \mathbf{r}_{j_1} + \boldsymbol{\omega}_j \times \left(\mathbf{r}_{j_2} - \mathbf{r}_{j_1}\right) + \mathbf{d}_{jk} \tag{6}$$

$$\boldsymbol{\omega}_k = \boldsymbol{\omega}_j + \boldsymbol{\omega}_{jk} \tag{7}$$

$$\mathbf{g}_k = \dot{\mathbf{r}}_{k_1} + \boldsymbol{\omega}_k \times \left(\mathbf{g}_k - \mathbf{r}_{k_1}\right) \tag{8}$$

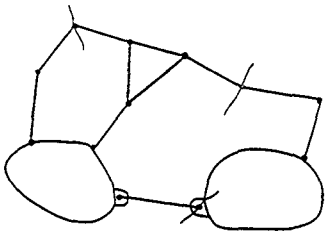


Figure 4. Opening a closed-loop system by cutting joints

$$r_{k_2} = g_k + \omega_k \times \left(r_{k_2} - g_k\right) \tag{9}$$

$$u_{k_1} = \omega_k \times u_{k_1} \tag{10}$$

The corresponding expressions for accelerations are

$$\ddot{r}_{k_1} = \ddot{r}_{j_1} + \dot{\omega}_j \times \left(r_{j_2} - r_{j_1}\right) + \omega_j \times \omega_j \times (r_{j_2} - r_{j_1}) + \ddot{d}_{jk} \tag{11}$$

$$\dot{\omega}_k = \dot{\omega}_j + \dot{\omega}_{jk} \tag{12}$$

$$g_k = \ddot{r}_{k_1} + \omega_k \times \left(g_k - r_{k_1}\right) + \omega_k \times \omega_k \times \left(g_k - r_{k_1}\right) \tag{13}$$

$$\ddot{r}_{k_2} = g_k + \omega_k \times \left(r_{k_2} - g_k\right) + \omega_k \times \omega_k \times \left(r_{k_2} - g_k\right) \tag{14}$$

$$u_{k_1} = \dot{\omega}_k \times u_{k_1} + \omega_k \times \omega_k \times u_{k_1} \tag{15}$$

These equations are particularized to the different joints in Appendix A

2 2. KINEMATICS OF CLOSED-CHAIN MULTIBODY SYSTEMS

Figure 4 shows a multibody system with several closed loops. This system can be transformed into one (or more) open-tree systems by "cutting" some of the joints. The effect of the cut joints is taken into account in the formulation by introducing the explicit constraint equations corresponding to these joints. Therefore, we will study a multibody system with closed loops as an open-loop system subject to the constraint equations of the cut joints.

Since we introduce some constraint equations, only a subset of all the joint coordinates are independent. Let F_o and F_c be the number of degrees of freedom of the open and closed loop systems, respectively (i e F_c is the actual number of degrees of freedom). If we call q the F_ox1

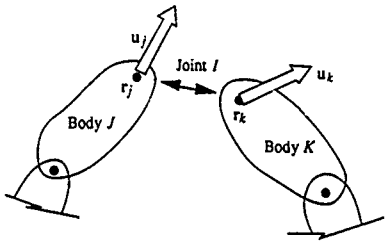


Figure 5. Generic joint model.

vector of open-loop joint variables, only a subset of those variables are independent. We call z the F_cx1 subset of q, that contains the independent joint coordinates.

To solve the position problem with complete generality, let us consider the establishment of the constraints corresponding to the generic joint I, located between bodies J and K, that is shown in figure 5. Each body has one point and one unit vector, and q_{jk} are the coordinates corresponding to the joint. The constraints are

$$\Phi^I(r_j, u_j, r_k, u_k) = 0 \tag{16}$$

Table 1 shows the constraint equations corresponding to the most common joints.

Joint	Constraints
Hinge	$r_j - r_k = 0$ $\quad u_j - u_k = 0$
Spherical	$r_j - r_k = 0$
Universal	$r_j - r_k = 0$ $\quad u_j^T u_k = 0$
Cylindrical	$u_j - u_k = 0$ $\quad (r_j - r_k) \times u_j = 0$
Prismatic	$u_j - u_k = 0$ $\quad (r_j - r_k) \times u_j = 0$ $\quad u_{j_2}^T u_{k_2} = 0$

Table 1. Constraint equations for cut joints

These equations must be formulated for all the cut joints to obtain the global system of nonlinear equations

$$\Phi(x(q)) = 0 \tag{17}$$

The vector of global positions x can be computed as a function of q, according to the expressions presented in section 2.1. The system of nonlinear equations (17) can be solved using the Newton-Raphson method, which can be written as

$$\Phi_q^{(i)} \Delta q^{(i+1)} = -\Phi^{(i)} \tag{18}$$

where the superscript indicates the iteration number. The iteration process must continue until the constraint equations are satisfied to a prescribed tolerance. Once obtained the solution q, x can easily be computed from the expressions derived in section 2.1 for open-loop systems. The derivation of the expressions for the jacobian Φ_q is illustrated in Appendix B through an example.

The velocity problem can be solved in two steps. First, calculate the dependent joint velocities $\dot{q}$ given the position vector x and the independent joint velocities $\dot{z}$. And second, calculate the Cartesian velocities of all the points and unit vectors using the open-chain multibody formulation explained in section 2.1. For the first step we need to differentiate all the constraint equations (17), that we assume holonomic and independent of time, to obtain

$$\Phi_q \dot{q} = 0 \tag{19}$$

The value of $\dot{q}$ is obtained from the solution of the system of linear equations (19) for the known values $\dot{z}$. The velocity of the remaining points and vectors can be calculated recursively from the expressions derived in section 2.1.

The acceleration analysis can be performed in a similar way. Differentiating expression (19), we obtain

$$\Phi_q \ddot{q} = -\Phi_q \dot{q} \tag{20}$$

Given x, $\dot{x}$ and the acceleration of the independent joint variables $\ddot{z}$, expression (20) represents a system of linear equations that can be solved for $\ddot{q}$. Once computed $\ddot{q}$, the accelerations of all the points and vectors comes from equations (11)-(15).

3. Dynamic Analysis

The virtual power form of the Newton-Euler equations of motion for a set of N_b interconnected rigid bodies is

$$Y^{*T}\left(MY + C - Q\right) = 0 \tag{21}$$

where M is the $6N_b$x$6N_b$ mass matrix, C is the $6N_b$x1 vector of centrifugal forces, Q is the $6N_b$x1 vector of applied external forces, Y is the $6N_b$x1 vector of Cartesian accelerations and Y^* is the $6N_b$x1 vector of virtual Cartesian velocities. These terms can be written as

$$M = \begin{bmatrix} M_1 & & \\ & \ddots & \\ & & M_{N_b} \end{bmatrix} \quad C = \begin{Bmatrix} C_1 \\ \vdots \\ C_{N_b} \end{Bmatrix} \quad Q = \begin{Bmatrix} Q_1 \\ \vdots \\ Q_{N_b} \end{Bmatrix} \quad Y = \begin{Bmatrix} Y_1 \\ \vdots \\ Y_{N_b} \end{Bmatrix} \tag{22}$$

The 6x6 mass matrix M_i and the 6x1vectors C_i, Q_i and Y_i corresponding to body i can be expressed as

$$M_i = \begin{bmatrix} m_i I & 0 \\ 0 & J_i \end{bmatrix}; \quad C_i = \begin{Bmatrix} 0 \\ \omega_i \times J_i \omega_i \end{Bmatrix} \quad Q_i = \begin{Bmatrix} f_i \\ n_i \end{Bmatrix} \quad Y_i = \begin{Bmatrix} \dot{g}_i \\ \omega_i \end{Bmatrix} \tag{23}$$

where m_i is the mass, J_i the 3x3 inertia tensor expressed in the inertial reference frame, I the 3x3 identity matrix, ω_i the angular velocity, g_i the position of the center of mass, f_i the 3x1vector of external forces and n_i the 3x1 vector of external torques applied at the center of mass. The value of J_i in terms of the constant inertia tensor of the body $\bar{J}_i$ in the body reference frame, is computed as

$$J_i = A_i \bar{J}_i A_i^T \tag{24}$$

Equation (21) is the virtual form of the equations of motion. In order to eliminate the virtual velocities from it, the Cartesian velocities Y must be expressed in terms of the independent joint variables z. This transformation from one set of velocities to the other is carried out in the next sections for open and closed loops.

3.1 DYNAMICS OF OPEN-LOOP SYSTEMS

Following Kim and Vanderploeg (1986) and assuming that the constraints are independent of time, we can write a linear relationship between the independent joint velocities $\dot{q}$ and the Cartesian velocities Y of the form

$$Y = R\dot{q} \tag{25}$$

where R is a velocity transformation matrix of size $6N_b xF$. Extracting the 6 rows of body i from expression (25), we obtain

$$Y_i = R_i \dot{q} \tag{26}$$

where R_i is a $6xF$ matrix. From expression (26) we deduce that the j-th column of R_i represents the body's Cartesian velocity when the j-th joint variable takes unit value and all the rest are zero. Therefore, the computation of R_i is equivalent to computing different sets of Cartesian velocities of the i-th body. The computation of R_i is presented in more detail in Appendix C.

Differentiating expression (26) we obtain

$$\dot{Y} = R\ddot{q} + \dot{R}\dot{q} \tag{27}$$

Extracting from equation (27) the 6 rows corresponding to body i, we can write

$$\dot{Y}_i = R_i \ddot{q} + (\dot{R}q)_i \tag{28}$$

which allows us to give the 6x1 vector $(\dot{R}q)_i$ a physical meaning. It represents the Cartesian acceleration vector of body i when all the joint accelerations are zero. Therefore, the computational cost of calculating $(\dot{R}q)_i$ is equivalent to an acceleration analysis.

Equation (27) and the virtual form of equation (25) can be substituted into (21), obtaining

$$R^T M R\ddot{q} = R^T (Q - C) - R^T M \dot{R}\dot{q} \tag{29}$$

which is the final form of the equations of motion for open-loop systems.

3.2 DYNAMICS OF CLOSED-LOOP SYSTEMS

Similarly to what was done in Section 2.2, a system with closed loops can be studied as a tree-structure system subject to constraints. To eliminate the closed loops one must "cut" one joint from each loop and then impose the cut-joint constraints explicitly.

Let F_o be the number of degrees of freedom of the tree-structure system, once the joints have been cut, and q the corresponding F_ox1 vector of joint coordinates. The components of q are not independent because they are related through the cut joint constraints. Let F_c be the number of the degrees of freedom of the closed loop system and z the F_cx1 vector of independent joint coordinates. A linear relationship between the two sets of joint velocities must exist and can be written as

$$\dot{\mathbf{q}} = \mathbf{R}_z \dot{\mathbf{z}} \tag{30}$$

where $\mathbf{R}_z$ is a F_oxF_c matrix. If the independent coordinates z are a subset of q, we can write

$$\dot{\mathbf{z}} = \mathbf{B}\mathbf{q} \tag{31}$$

where B is a F_cxF_o constant matrix composed of zeros and ones.

The matrix $\mathbf{R}_z$ can then easily be computed as follows by appending equation (31) to (19), which leads to

$$\begin{bmatrix} \Phi_q \\ \mathbf{B} \end{bmatrix} \dot{\mathbf{q}} = \begin{Bmatrix} 0 \\ \dot{\mathbf{z}} \end{Bmatrix} \tag{32}$$

Isolating $\dot{\mathbf{q}}$ we finally obtain

$$\dot{\mathbf{q}} = \begin{bmatrix} \Phi_q \\ \mathbf{B} \end{bmatrix}^{-1} \begin{Bmatrix} 0 \\ \dot{\mathbf{z}} \end{Bmatrix} = \mathbf{R}_z \dot{\mathbf{z}} \tag{33}$$

which tells us that $\mathbf{R}_z$ can be computed as the last F_c columns of the inverse matrix. Differentiating equation (33) we find

$$\mathbf{q} = \mathbf{R}_z \ddot{\mathbf{z}} + \dot{\mathbf{R}}_z \mathbf{z} \tag{34}$$

Substituting equation (34) into the equations of motion (29) and premultiplying by $\mathbf{R}_z^T$ we obtain

$$\overline{\mathbf{R}}^T \mathbf{M} \overline{\mathbf{R}} \ddot{\mathbf{z}} = \overline{\mathbf{R}}^T (\mathbf{Q} - \mathbf{C}) - \overline{\mathbf{R}}^T \left(\mathbf{M} \dot{\mathbf{R}} \dot{\mathbf{q}} + \mathbf{R} \dot{\mathbf{R}}_z \dot{\mathbf{z}} \right) \tag{35}$$

where $\overline{\mathbf{R}}$ is the $6N_b$xF_c matrix

$$\overline{\mathbf{R}} = \mathbf{R}\mathbf{R}_z \tag{36}$$

Expression (36) constitutes the set of F_c second-order differential equations of motion of the closed-loop system

2.3 ALGORITHM FOR DYNAMIC SIMULATION

The aim of this section is to present a general algorithm for dynamic analysis of open and closed-loop systems that takes into account several efficiency issues. The algorithm is as follows.

1. Define the topology of the multibody system and label all the bodies, joints, points, vectors and joint variables
2. Select F_c independent joint variables and assign values to the position $\mathbf{z}$ and the velocity $\dot{\mathbf{z}}$ at time $t=0$.
3. Calculate $\Phi_{\mathbf{q}}$ and solve iteratively equation (18) for $\mathbf{q}$ and (1)-(5) for $\mathbf{x}$.
4. Solve the velocity problem through equation (19) to calculate $\dot{\mathbf{q}}$ and (6)-(10) to calculate $\dot{\mathbf{z}}$.
5. Compute the open-loop velocity-dependent accelerations $\mathbf{R}\dot{\mathbf{q}}$.
6. Solve the linear system of equations (33) to obtain the matrix $\mathbf{R}_{\mathbf{z}}$. Calculate $\mathbf{R}_{\mathbf{z}}\dot{\mathbf{z}}$.
7. *DO PARALLEL* $i=1, N_b$
   - 7.1 Calculate the terms $\mathbf{M}_i$, $\mathbf{C}_i$, $\mathbf{Q}_i$ and $\mathbf{R}_i$
   - 7.2 Calculate $\overline{\mathbf{R}}_i$ by multiplying $\mathbf{R}_i \mathbf{R}_{\mathbf{z}_i}$.
   - 7.3 Multiply $\overline{\mathbf{R}}_i^T \mathbf{M}_i$, $\overline{\mathbf{R}}_i^T (\mathbf{Q}_i - \mathbf{C}_i)$ and $\overline{\mathbf{R}}_i^T (\dot{\mathbf{R}}_{\mathbf{z}} \dot{\mathbf{z}})$.
   - 7.4 Multiply $(\overline{\mathbf{R}}_i^T \mathbf{M}_i)(\mathbf{R}\dot{\mathbf{q}})$ and $\overline{\mathbf{R}}_i^T (\mathbf{M}_i \overline{\mathbf{R}}_i)$.
8. Factorize $\overline{\mathbf{R}}^T \mathbf{M} \overline{\mathbf{R}}$ in equation (35) and solve for $\dot{\mathbf{z}}$.
9. Integrate $(\dot{\mathbf{z}}, \dot{\mathbf{z}})_t$ to obtain $(\mathbf{z}, \dot{\mathbf{z}})_{t+\Delta t}$
10. Set $t = t+\Delta t$ and go to step 3.

The *DO PARALLEL* statement used in point 7 represents the system-dependent directive for parallelization. This directive splits the iteration loop into subtasks and sends one or more subtasks to each processor.

The most attractive feature of this algorithm is that there is a significant portion of the code that is subject to run in parallel. The parallel loop contains one iteration per body, which allows for the body by body computation of the final vectors and matrices of equation (35). For moderate size problems, the typical amount of code (measured in terms of floating point operations) that may run in parallel is between 50 and 70%. To take full advantage of the power of parallel computation it is necessary to balance the amount of work sent to each processor. The goal is to have the job evenly distributed among processors and to avoid having some idle processors while others are overloaded.

An efficient implementation of the matrix products requires that the sparsity pattern of the matrices be considered. The matrices usually have wide portions of zeros that can be exploited to speed up the computations. Furthermore, the matrix $\overline{\mathbf{R}}^T \mathbf{M} \overline{\mathbf{R}}$ may have regions of zeros that can be considered by using a skyline profile solver

4. Examples

4.1 5-BAR PENDULUM

Figure 6 shows a 5-bar pendulum with 5 revolute joints and 5 degrees of freedom. The length of all the links is $L = 1$ m and they have a uniformly distributed mass $M = 1$ Kg. The pendulum, initially at rest, is left free and evolves under the effect of gravity $g = 9.81$ m/s^2. The equations of motion were integrated for 10 s using Shampine and Gordon's (1975) DE integrator with an error tolerance of 10^{-5}. Figure 8 shows the x, y and z coordinates of the point P, shown in figure 6.

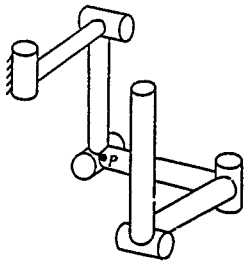


Figure 6. Spatial 5-bar pendulum

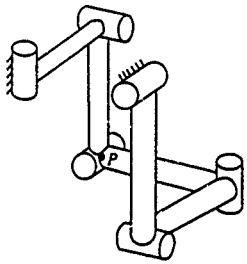


Figure 7 Bricard mechanism

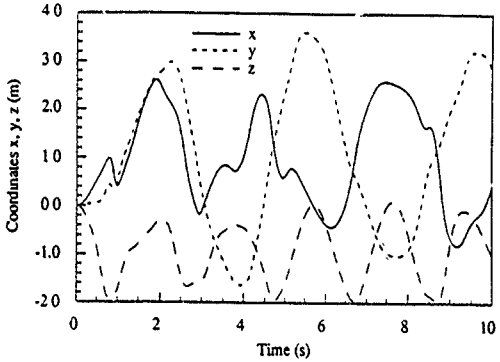


Figure 8 Coordinates of point *P* of the 5-bar pendulum

The CPU times for this example obtained with the multibody program COMPAMM and the proposed method are summarized in table 2.

4.2 BRICARD MECHANISM

The single degree of freedom closed-loop mechanism of figure 7 is called *Bricard mechanism*. It is essentially identical to the 5-bar pendulum seen in the previous section, except that the last body is hinged to the ground. All the geometrical and inertial properties remain unchanged from the previous section. The mechanism is initially at rest. It is left free and evolves under the effect of gravity $g = 9.81$ m/s^2. The equations of motion are integrated for 15 s with the same error

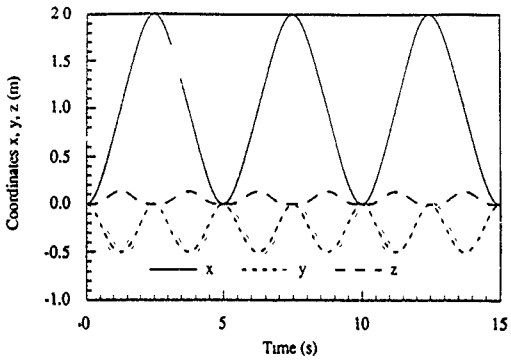


Figure 9. Coordinates of point P of the Bricard mechanism

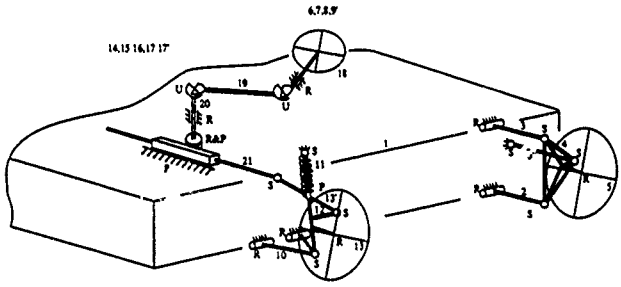


Figure 10. Left half of a full car model.

tolerance. Figure 9 shows the x, y, z coordinates of the point P shown in figure 7. A comparison between the CPU times obtained with the program COMPAMM and with the real-time method are shown in table 2.

4.3 FULL CAR MODEL

Figure 10 represents the left half of a car model that consists of the steering mechanism and the front and rear suspensions The full model (comprising the right half not shown) consists of 25 rigid bodies and has 15 degrees of freedom, 6 of them corresponding to the rigid body motion of the chassis, 4 to the up and down motion of each suspension, 4 to the wheel rotations and one to

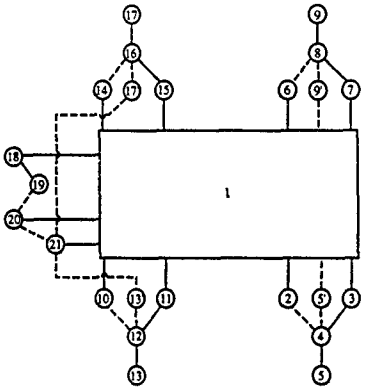


Figure 11. Topological graph of the full car model

the rotation of the steering wheel. The joint types are indicated in the figure by a character: *R* means *Revolute*, *U* means *Universal*, *S* means *Spherical*, *P* means *Prismatic* and *R&P* means *Rack and Pinion*. The numbering of the bodies corresponding to the right front and rear suspensions is indicated on the figure.

The car model contains several closed loops. To visualize the closed loops in a clear way, a schematic graph is shown in Figure 11. The nodes are the bodies and the edges are the joints. To turn the car model into an open-loop mechanism it is necessary to remove some joints or bodies. The edges depicted in dashed lines correspond to the cut joints. The solid lines indicate the tree-structure system. Instead of removing the 8 joints corresponding to the bodies 5', 9', 13' and 17', the 4 bodies themselves have been removed, which has the advantage of reducing the size of the open-loop system. The mass of the removed bodies has been transmitted to the adjacent bodies so that the formulation is consistent.

The contact between the tire and the ground has been modeled with the Calspan tire force model (Bohn and Keenan, 1974)

The CPU times required for this simulation are presented in table 2.

4 4 CAR CRASH SIMULATION WITH HUMAN BODY MODEL

Figure 12 represents a human body model with 45 degrees of freedom. The degrees of freedom are distributed as follows: 3 at the neck, 2 at each collar bone, 3 at each shoulder, 2 at each elbow, 2 at each wrist, 1 for finger motion on each hand, 2 at the waist, 3 at each hip, 1 at each knee and 3 at each foot. In addition, there are 6 degrees of freedom of the base body, which is taken to be the hips. The 3, 2 and 1 degrees of freedom motions at the joints have been represented with spherical, universal and revolute joints, respectively. For convenience, the

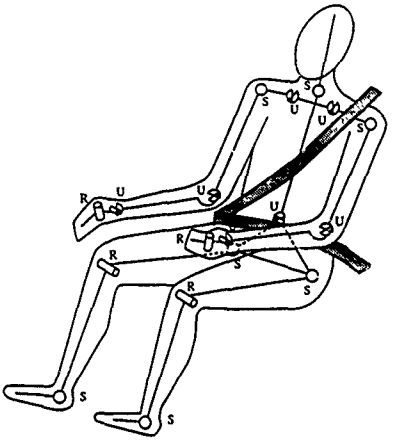


Figure 12. Car crash simulation with a 45 degrees of freedom model of the driver

model used for computation used only revolute joints. Universal joints were represented as two perpendicular revolute joints and spherical joints as three mutually perpendicular revolute joints.

The driver is assumed to be seated in a natural driving position. To make the driver maintain the driving position, torsional springs have been introduced at the joints. Before the frontal collision takes place, the driver is traveling at a constant speed of 5 m/s (18 Km/h). The effect of the seat belt has been modeled as a force element that restrains the motion of the chest and hips.

Figure 13 shows the acceleration plot of the driver's neck. CPU times are shown in table 2.

Example	COMPAMM	Real-time	Ratio
5-bar pendulum	17 6	1.1	16 0
Bricard mechanism	8 2	1.1	7.5
Full car model	203 5	10 2	20 0
Car crash simulation	785 4	14 2	55 3

Table 2. CPU times per function evaluation measured in milliseconds on a single processor of a SGI 4D/240 workstation.

References

Armstrong, W. W , 1979, "Recursive Solution to the Equations of Motion of an N-Link Manipulator', *5th World Congress on Theory of Machines and Mechanisms*, vol. 2, pp. 1343-1346

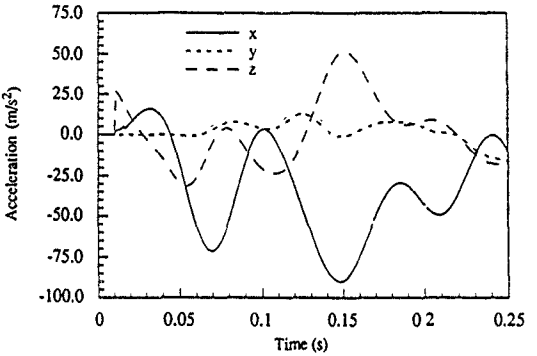


Figure 13. Acceleration of the driver's neck.

Avello, A., Jiménez, J. M., Bayo, E. and García de Jalón, J., 1993, "A Simple and Highly Parallelizable Method for Real-Time Dynamic Simulation Based on Velocity Transformations", To Appear in *Computer Methods for Applied Mechanics and Engineering*.

Bae, D. S. and Haug, E J., 1987, "A Recursive Formulation for Constrained Mechanical System Dynamics. Part I: Open Loop Systems', *Mechanical Structures and Machines*, vol. 15 (3), pp. 359-382.

Bae, D. S., Kuhl, J. G. and Haug, E. J., 1988, "A Recursive Formulation for Constrained Mechanical System Dynamics: Part III. Parallel Processor Implementation", *Mechanical Structures and Machines*, vol. 16 (2), pp. 249-269.

Bohn, P. F. and Keenan, R. J., "Hybrid Computer Vehicle Handling Program", APL, The Johns Hopkins University, Report No. DOT-HS-801 290.

Chang J. L. and Kim, S. S , 1989, "A Low-Cost Real-Time Man-in-the-Loop Simulation for Multibody Systems", *Advances in Design Automation*, vol. 3, pp. 95-99

Featherstone, R., 1987, *Robot Dynamics Algorithms*, Kluwer Academic Publishers

Fijany, A. and Bejczy, A. K., 1990, "An Efficient Method for Computation of Manipulator Inertia Matrix", *Journal of Robotic Systems*, vol 7 (1), pp. 57-80.

García de Jalón, J., Jiménez, J. M., Martín, F. and Cuadrado, J., 1989, "Real-time simulation of complex 3-D multibody systems with realistic graphics", in *Real-Time Integration Methods for Mechanical System Simulation*, Ed. Haug, E. J. and Deyo, R. D., NATO ASI Series, Springer-Verlag.

Huston , R. L , 1990, "Multibody Dynamics", Butterworth-Heinemann

Jain, A, 1991, "Unified Formulation of Dynamics for Serial Rigid Multibody Systems", *Journal of Guidance, Control and Dynamics*, vol. 14 (3), pp 531-542.

Jerkovsky, W., 1978, "The Structure of Multibody Dynamics Equations", *Journal of Guidance and Control*, vol. 1 (3), pp 173-182.

Kane, T. R. and Levinson, D. A., 1985, *Dynamics: Theory and Applications*, McGraw-Hill.
Kim, S. S. and Vanderploeg, M. J., 1986, "A General and Efficient Method for Dynamic Analysis of Multibody Systems Using Velocity Transformations", *Journal of Mechanisms, Transmissions and Automation in Design*, vol 108, pp. 176-182.
Rosenthal, D, 1987, "Order N Formulation for Equations of Motion of Multibody Systems", *SDIO/NASA Workshop on Multibody Simulations*, JPL Pub D-5190, Jet Propulsion Laboratory, Pasadena, CA, pp. 1122-1150.
Schiehlen, W. O., 1990, *Multibody Systems Handbook*, Springer-Verlag.
Shampine, L. F and Gordon, 1975, *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, Freeman.
Walker, M. W. and Orin, 1982, "Efficient Dynamic Computer Simulation of Robotic Mechanisms", *ASME Journal of Dynamic Systems and Measurement Control*, vol. 104, pp. 205-211.

Appendix A

This appendix particularizes the expressions (1)-(15) for the most common types of joints

- *Hinge*: points r_{j_1} and r_{k_1}, and vectors u_{j_2}, u_{k_1} coincide respectively The joint coordinate z_{jk} is an angle defined on the unit vector u_{j_2}.

$$d_{jk} = \dot{d}_{jk} = \bar{d}_{jk} = 0 \tag{A.1}$$

$$A_{jk} = A_{jk}(z_{jk}, \bar{u}_{j_2}) = I_3 + \bar{u}_{j_2} \cdot \bar{u}_{j_2}(1 - \cos z_{jk}) + \bar{u}_{j_2} \sin z_{jk} \tag{A.2}$$

$$\omega_{jk} = z_{jk} u_{j_2} \tag{A.3}$$

$$\dot{\omega}_{jk} = z_{jk} u_{j_2} + z_{jk} \omega_j \times u_{j_2} \tag{A.4}$$

where the upper tilde denotes the 3x3 skew-symmetric matrix corresponding to a cross product.

- *Prismatic*: d_{jk} has the same direction as vectors u_{j_2} and u_{k_1}, and its module is defined by the joint coordinate z_{jk}

$$d_{jk} = z_{jk} u_{j_2} \tag{A.5}$$

$$\dot{d}_{jk} = z_{jk} u_{j_2} + z_{jk} \omega_j \times u_{j_2} \tag{A.6}$$

$$\ddot{d}_{jk} = \ddot{z}_{jk} u_{j_2} + 2 z_{jk} \omega_j \times u_{j_2} + z_{jk} \dot{\omega} \times u_{j_2} + z_{jk} \omega_j \times \left(\omega_j \times u_{j_2} \right) \tag{A.7}$$

$$A_k = A_j \tag{A.8}$$

$$\omega_{jk} = \omega_{jk} = 0 \tag{A.9}$$

- *Cylindrical:* it is similar to the revolute joint, but with an additional rotation along the axis defined by the vector u_{j_2}. If z^t_{jk} and z^r_{jk} are the joint coordinates of translation and rotation respectively

$$d_{jk} = z^t_{jk} u_{j_2} \tag{A.10}$$

$$A_k = A_j A_{jk}(z^r_{jk}, \bar{u}_{j_2}) \tag{A.11}$$

$$\dot{d}_{jk} = z_{jk} u_j + z^t_{jk} \omega_j \times u_{j_2} \tag{A.12}$$

$$\omega_{jk} = z^r_{jk} u_{j_2} \tag{A.13}$$

$$\ddot{d}_{jk} = z^t_{jk} u_{j_2} + 2 z^t_{jk} \omega_j \times u_{j_2} + z^t_{jk} \omega_j \times u_{j_2} + z^t_{jk} \omega_j \times (\omega_j \times u_{j_2}) \tag{A.14}$$

$$\omega_{jk} = z^r_{jk} u_{j_2} + z^r_{jk} \omega_j \times u_{j_2} \tag{A.15}$$

- *Universal:* the points r_{j_2} and r_{k_1} coincide If z^1_{jk} and z^2_{jk} are the two rotations about the two axis defined by the vectors u_{j_2} and u_{k_1} respectively, then

$$d_{ij} = d_{jk} = \dot{d}_{jk} = 0 \tag{A.16}$$

$$A_k = A_j A^1_{jk}(z^1_{jk}, \bar{u}_{j_2}) A^2_{jk}(z^2_{jk}, \bar{u}_{k_1}) \tag{A.17}$$

$$u_{k_1} = A_k \bar{u}_{k_1} \tag{A.18}$$

$$\dot{u}_{k_1} = (\omega_j + z^1_{jk} u_{j_2}) \times u_{k_1} \tag{A.19}$$

$$\omega_k = \omega_j + z^1_{jk} u_{j_2} + z^2_{jk} u_{k_1} \tag{A.20}$$

$$\omega_k = \omega_j + z^1_{jk} u_{j_2} + z^2_{jk} u_{k_1} + z^1_{jk} u_{j_2} + z^2_{jk} u_{k_1} \tag{A.21}$$

- *Spherical:* the points r_{j_2} and r_{k_1} coincide. Using the Euler Parameters $p \equiv (e_0, e_1, e_2, e_3)$ (which must satisfy the constraint $e_0^2 + e_1^2 + e_2^2 + e_3^2 = 1$) to define the three possible rotations

$$d_{ij} = \dot{d}_{jk} = d_{jk} = 0 \tag{A.22}$$

$$A_k = A_j A_{jk}(z_{jk} \equiv p) \tag{A.23}$$

$$A_{jk}(p) = (2e_0^2 - 1) I_3 + 2(ee^T + e_0 \tilde{e}) \tag{A.24}$$

$$\omega_{jk} = 2Gp \tag{A.25}$$

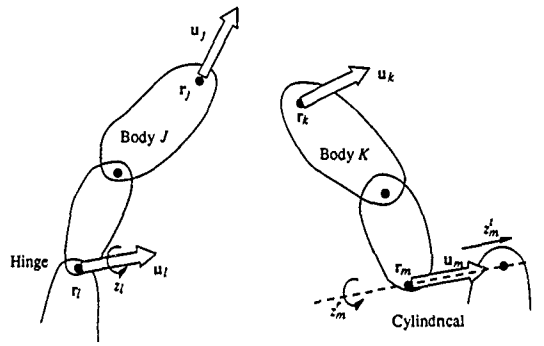


Figure 14 Calculation of the derivatives of a loop closure condition.

$$G \equiv \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \tag{A.26}$$

$$\omega_{jk} = 2Gp \tag{A.27}$$

- *Floating*: the three translations are defined by the joint coordinates z^1_{jk}, z^2_{jk} and z^3_{jk}, that coincide with the coordinates of point r_k with respect to a reference frame (m, n, l) parallel to the inertial ones with its origin in the point r_{j_2}. The three rotations are introduced in a way similar to the spherical joint

$$d_{jk} = z^1_{jk}m + z^2_{jk}n + z^3_{jk}l \tag{A.28}$$

$$\dot{d}_{jk} = z^1_{jk}m + z^2_{jk}n + \dot{z}^3_{jk}l \tag{A.29}$$

$$d_{jk} = z^1_{jk}m + z^2_{jk}n + z^3_{jk}l \tag{A.30}$$

Appendix B

The jacobian Φ_q of the constraint equations shown in table 1 can be calculated as

$$\Phi_q = \Phi_x x_q \tag{B.1}$$

The matrix Φ_x is derived in a straightforward way by inspection of table 1. To calculate Φ_q, it is necessary to evaluate the derivatives of the points and vectors with respect to the joint coordinates q. This needs to be done only once for each type of joint since the results may be stored in a joint library and used for all subsequent joints of the same type. Observe that

$$\frac{\partial \mathbf{r}_j}{\partial q_k} = \frac{\partial \dot{\mathbf{r}}_j}{\partial \dot{q}_k} = \mathbf{v}_{jk} \tag{B.2}$$

which means that the terms in Φ_q are partial velocities (see Kane and Levinson, 1985) that may be computed by calculating the velocity $\dot{\mathbf{r}}_j$ corresponding to an input zero velocity for all the joint variables except for $\dot{q}_k = 1$.

Let us illustrate this with the example shown in figure 14. The figure shows a cut joint l between bodies J and K. There is a hinge joint with a joint angle z_l on the left branch and a cylindrical joint with a joint angle z_m^r and a joint offset z_m^t on the right branch. The partial derivatives mentioned above are shown in table 3.

	$\partial \mathbf{r}_l$	$\partial \mathbf{u}_j$	$\partial \mathbf{r}_k$	$\partial \mathbf{u}_k$
∂z_l	$\mathbf{u}_l \times (\mathbf{r}_j - \mathbf{r}_l)$	$\mathbf{u}_l \times \mathbf{u}_j$	0	0
∂z_m^r	0	0	$\mathbf{u}_m \times (\mathbf{r}_k - \mathbf{r}_m)$	$\mathbf{u}_m \times \mathbf{u}_k$
∂z_m^t	0	0	$\mathbf{u}_m$	0

Table 3. Jacobian Φ_q for the example of figure 14

Appendix C

In this Appendix we present the expressions for the calculation of the submatrices $\mathbf{R}_i$ of equation (26). Each submatrix $\mathbf{R}_i$ is of size $6xF_o$, where F_o is the number of degrees of freedom of the open-loop system. Each column of $\mathbf{R}_i$ is computed by giving a unit velocity to each joint variable and by computing the resulting body's velocity. If the body is on a different branch than the unit joint velocity, the corresponding column of $\mathbf{R}_i$ is zero. Likewise, the columns of $\mathbf{R}_i$ corresponding to the joint variables located further along a forward path are also zero. The only nonzero columns of $\mathbf{R}_i$ are those of the joint variables preceding the body on the same forward path.

We now present the values of $\mathbf{R}_i$ for the joint variables corresponding to the most common types of joints. The number of columns, that varies from one joint to the other depending on the number of degrees of freedom, is shown on the lower right part of the matrix.

- *Hinge* Defined by point $\mathbf{r}_j$ and vector $\mathbf{u}_j$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{u}_j \times (\mathbf{g}_i - \mathbf{r}_j) \\ \mathbf{u}_j \end{bmatrix}_{6x1} \tag{C.1}$$

- *Prismatic*: Defined by vector $\mathbf{u}_j$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{u}_j \\ 0 \end{bmatrix}_{6x1} \tag{C.2}$$

- *Cylindrical*. Defined by point $\mathbf{r}_j$ and vector $\mathbf{u}_j$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{u}_j \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{u}_j \\ \mathbf{u}_j & \mathbf{0} \end{bmatrix}_{6x2} \tag{C 3}$$

- *Universal*. Defined by point $\mathbf{r}_j$ and vectors $\mathbf{u}_j$ and $\mathbf{u}_k$.

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{u}_j \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{u}_k \times (\mathbf{g}_i - \mathbf{r}_j) \\ \mathbf{u}_j & \mathbf{u}_k \end{bmatrix}_{6x2} \tag{C.4}$$

- *Spherical*: Defined by point $\mathbf{r}_j$.

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{i} \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{j} \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{k} \times (\mathbf{g}_i - \mathbf{r}_j) \\ \mathbf{i} & \mathbf{j} & \mathbf{k} \end{bmatrix}_{6x3} \tag{C 5}$$

where **i**, **j** and **k** are unit vectors pointing in the direction of the x, y and z inertial axes, respectively

- *Floating*: Defined by point $\mathbf{r}_j$

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} & \mathbf{i} \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{j} \times (\mathbf{g}_i - \mathbf{r}_j) & \mathbf{k} \times (\mathbf{g}_i - \mathbf{r}_j) \\ 0 & 0 & 0 & \mathbf{i} & \mathbf{j} & \mathbf{k} \end{bmatrix}_{6x6} \tag{C 6}$$

Efficient Object-Oriented Programming of Multibody Dynamics Formalisms

U. REIN
Institut A für Mechanik
Universität Stuttgart
Pfaffenwaldring 9
W-7000 Stuttgart 80
Germany

ABSTRACT. A class structure for the object-oriented, numerical language C++ is presented to replace the built-in scalar type *double* with a new scalar class *se* (scalar environment) Matrix- and vector-classes programmed with the use of the scalar class *se* allow comfortable object-oriented programming and modifying of multibody formalisms. Because an instance of a *se* stores further information besides the magnitude of a scalar number, operations with *se's* can be optimized with respect to multibody formalisms. The number of floating point operations remaining after optimization can be counted and an executable program is created, which executes only these minimum number of floating point operations. This serves as a basis for comparing, modifying and optimizing given formalisms with respect to the multibody system under investigation. As an example, an implementation of the recursive formalism is presented in the paper.

1. Introduction

During the last decades a large number of multibody dynamics formalisms for different purposes have been published. Especially the development of the recursive formalism [1], [18], [7], [4] has shown the need for comparing the efficiency of multibody dynamics formalisms. This has been done using different methods, e g. counting the necessary operations by hand [4], [15], with the aid of a symbolic manipulation program [16], or by measuring the CPU-time for one evaluation of the right-hand side [13]. For a serial chain of rigid bodies connected by revolute joints, Fig. 1 shows the basic difference between an implementation of the recursive and an implementation of the nonrecursive method described in [12]. While the CPU-time of the recursive formalism increases with Order(n), i.e. linearly with the number of joints n, the nonrecursive method increases with Order(n^3). The intersection point is at about five joints. Surprisingly, this doesn't hold any more for systems with parallel tree structure, i.e. sytems where the number of branches is close to the number of bodies, since the nonrecursive method used here for the comparison [12] also shows an almost linear behaviour. Thus, for many practical applications an optimized nonrecursive method is faster than the recursive method.

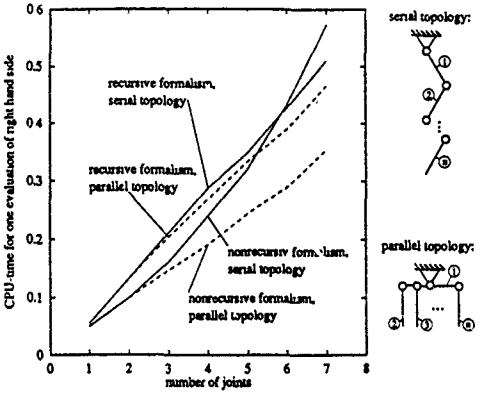


Figure 1: Comparison of the recursive with a nonrecursive formalism [13]

There were published a number of versions of the recursive formalism [7], [4], [2], [17]. They use different coordinate systems and reference points to evaluate the recursive formalism. These versions of the recursive formalism have been compared in [15] and [16], using again a serial chain of rigid bodies, connected by revolute joints. Fig. 2 shows the differences between the versions of the recursive formalism.

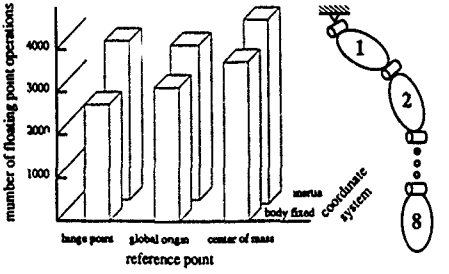


Figure 2: Comparison of variants of the recursive formalism for a serial chain with revolute joints [15]

The minimum number of floating point operations are based on very special optimizations, which are only possible for the benchmark system used, i.e. a serial chain with rigid bodies connected by revolute joints. These well-known results show that a multibody system program can be made more efficient by choosing an adequate formalism. As Fig 1 shows, the efficiency of a formalism depends strongly on the system it is applied to. Unfortunately, there are only a few comparisons for some formalisms on only some benchmark problems. To find an optimal formalism for a given problem, it would therefore be helpful to have a better comparison of formalisms on a wider variety of systems and to have an environment to program and modify comfortably formalisms and to apply them to a particular system.

One answer of computer science to program and modify quickly and savely large and complex algorithms like multibody dynamics formalisms is to use object oriented programming (OOP) [3]. There is a large potential to do this in multibody dynamics on different levels:

- OOP on the system level: description of the given system with objects (bodies, joints, coordinate systems, subsystems etc.),
- OOP on the formalism level: defining kinematic and kinetic methods for the objects introduced on the system level,
- OOP on the linear algebra level: matrix and vector operations,
- OOP on the scalar level.

Since there are already objects in multibody systems, OOP on the system level has always been done. Recent investigations consequently use object-oriented models for the database to store the system information [11]. Using OOP on the formalism level is more difficult and a current field of research [9], [10]. OOP on the linear algebra level is quite common, there are already commercial packages available. There are several methods to do OOP on the scalar level:

- Using the built-in scalar types like *double* in C++. This is inefficient. For example, when programming the recursive formalism using vector and matrix notation, the numbers of Fig. 2 can only be achieved by taking care of zeros, ones, constants, as well as unnecessary and equal operations and so on.
- Using a symbolic manipulator for scalar arithmetic and an OOP-language for the rest. This is uncomfortable to program because of the two programming packages.
- Using special objects to do scalar arithmetic optimized for multibody formalisms. This has been done e.g. by SAYER in the program AUTOSIM [14]. Since AUTOSIM is encoded in the numerically less efficient language LISP, this requires again another compiler to run the final program.

The language C++ claims to be an efficient numerical language and to allow object oriented programming [6]. This paper describes a way to do optimized scalar arithmetic in C++. An algorithm programmed with newly introduced scalar objects will be automatically optimized with respect to a multibody dynamics formalism programmed in vector and matrix notation. The paper describes first the programming concept. Second, as an example, an implementation of the recursive formalism in the form suggested by BRANDL/JOHANNI/OTTER [4] is described.

2. Object oriented environment for optimized scalar algebra in C++

The scalar types *float* or *double* of C++ don't take care of special information which would be necessary to optimize a multibody dynamics formalism programmed in matrix and vector notation. Even a standard optimizer isn't capable to do this to a full extend. Important optimization information of scalars are

- the origin of a number: constant from file, internal constant, internal variable, input variable from integrator, integration variable,
- special constants like *zero* or *one*,
- the history of operations, which are necessary to calculate a number.

Therefore, a new class *se* has been developed, which completely replaces the build-in scalar type *double* of C++. An instance of *se* references not any more a number itself but instead all information to optimize operations formulated with *se's*. All standard operators have been overloaded, so there is almost no difference between programming with *doubles* and programming with *se's*. One advantage of using overloaded operators is that there is no need to write a scanner and parser as in a symbolic language, since the scanner and parts of the parser from C++ itself is used [8]. The basic class structure is taken from a simplified model of a macro-assembler or compiler [8], see Fig. 3.

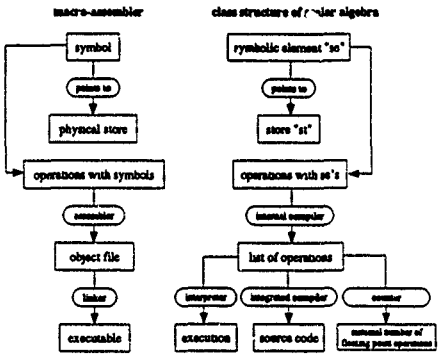


Figure 3: Class structure of the scalar programming environment

The result of running a program formulated with the scalar objects *se* will be a list of operations, which can be carried out immediately by interpreting it or with full speed after another compilation step. The main advantage of the list of operations together with the special information stored for each scalar is the possibility to perform local and global optimizations.

- Local optimizations result immediately from one operation and its operands, e.g $0 \cdot x \mapsto 0$, $1 \cdot x \mapsto x$, or $\frac{a}{b} \cdot b \mapsto a$, ...
- Global optimizations result from information about the complete list of all operations. At the moment
  - elimination of equal operations,
  - elimination of dead end code,
  - separation of operations with constant results, and
  - separation of operations only used for report

  is implemented.

To simulate the forward dynamics of a multibody system, an integrator-function calls very often the right hand side of the equations of motion. At best, the whole right hand side is programmed using the newly introduced scalar class *se*. Usually less often than the integration variables, report variables are calculated at desired timesteps. The scalar class *se* distinguishes between an integrator variable and a report variable. All operations to calculate only report variables are separated and carried out only if necessary. There has to be an interface between the part of the program written in convential C/C++, e.g. the integrator, and the algorithm, which is programmed using the scalar class *se*. This is done with the interface variables for input, output, and report which are *se's* and global variables at the same time (see Fig. 4).

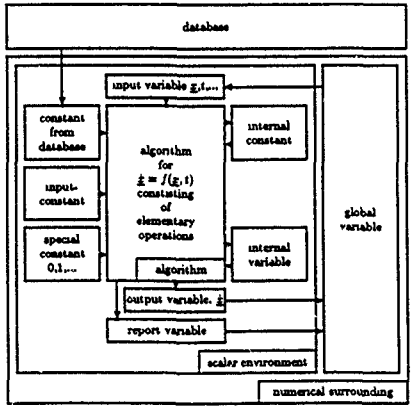


Figure 4: Interface of scalar environment to sorrounding program

After the list of operations has been optimized, the operations can be counted. Thus, the result is the minimum number of floating point operations together with an executable program which runs with this minimum number of operations.

3. Example

The previously introduced method is now applied to a benchmark problem. The forward dynamics of a serial chain of n rigid bodies, connected by revolute joints, is solved with the recursive method, see Fig. 5.

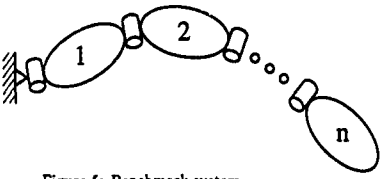


Figure 5: Benchmark system

The recursive method can be derived from the locally formulated equations of motion [17]. Sixdimensional notation will be used, where the 3×1-velocity v and the 3×1-angular velocity ω of a rigid body are combined in the 6×1 velocity $v = (v^T, \omega^T)^T$. Similarly a 3×1-force ν and a 3×1-moment μ acting on a body are combined to a 6×1-force $f = (\nu^T, \mu^T)^T$. With the symmetric, positive definite 6×6 mass matrix M, the Jacobian H, the joint variables θ, the sixdimensional joint reaction force f and the sixdimensional force g, which collects the external, applied forces and moments and the gyroscopic terms, the equations of motion are

$$\begin{aligned} M_j \dot{v}_j - f_j + {}_jG^{k^T} f_k &= g_j \quad , \quad j = 1, \ldots, n \\ \dot{v}_j - {}_jG^i \dot{v}_i - H_j \ddot{\theta}_j &= \gamma_j \quad , \quad j = 1, \ldots, n \\ H_j^T f_j &= 0 \quad , \quad j = 1, \ldots, n \quad . \end{aligned}$$

The index "j"denotes an arbitrary body in the chain with a possible predecessor, indexed with "i", and a possible successor, indexed with "k". The matrix

$${}_jG^i = \begin{pmatrix} I & \tilde{d}_{ji} \\ 0 & I \end{pmatrix}$$

transforms the sixdimensional velocity of body "i", ${}_iv_i$, with respect to reference point O_i to the same velocity of body "i", ${}_jv_i$, but with respect to reference point O_j, where the distance from O_j to O_i is d_{ji}. To simplify notation, the left lower index for the reference point is omitted, if identical with the right lower index for the body. These equations of motion can be solved recursively [4], [17], see Fig. 6.

- Projection: for $j = n, \ldots, 2$ do

$$
\begin{aligned}
M_i &= M_i + {}_jG^T(M_j - M_jH_j(H_j^TM_jH_j)^{-1}H_j^TM_j){}_jG \\
g_i &= g_i + {}_jG^T(I - M_jH_j(H_j^TM_jH_j)^{-1}H_j^T)(g_j - M_j\gamma_j)
\end{aligned}
$$

- Solving for root.

$$
\begin{aligned}
\ddot{\theta}_1 &= (H_1^TM_1H_1)^{-1}H_1^T(g_1 - M_1\gamma_1) \\
v_1 &= H_1\ddot{\theta}_1 + \gamma_1
\end{aligned}
$$

- Backsubstitution for $j = 2, \ldots, n$ do

$$
\begin{aligned}
\ddot{\theta}_j &= (H_j^TM_jH_j)^{-1}H_j^T(g_j - M_j({}_jG^Tv_i + \gamma_j)) \\
v_j &= H_j\ddot{\theta}_j + {}_jG^Tv_i + \gamma_j
\end{aligned}
$$

Figure 6: Recursive solution of equations of motion [17]

A very efficient computational scheme to solve the equations from Fig. 6 is given in [15], see Fig. 7.

- Projection: for $j = n \ldots 2$ do

$$
\begin{aligned}
V_j &= M_jH_j \;,\; D_j = (H_j^TV_j)^{-1} \\
F_j &= V_jD_j \;,\; w_j = H_j^Tg_j \\
M_j^P &= M_j - F_jV_j^T \\
g_j^P &= M_j^P\gamma_j + F_jw_j - g_j \\
M_i &= M_i + {}_jG^TM_j^P{}_jG \\
g_i &= g_i - {}_jG^Tg_j^P
\end{aligned}
$$

- Solving for root for $j = 1$ do

$$
\begin{aligned}
V_1 &= M_1H_1 \;;\; D_1 = (H_1^TV_1)^{-1} \\
F_1 &= V_1D_1 \;,\; w_1 = H_1^Tg_1 \\
\ddot{\theta}_1 &= D_1w_1 - F_1^T\gamma_1 \\
v_1 &= \gamma_1 + H_1\ddot{\theta}_1
\end{aligned}
$$

- Backsubstitution: for $j = 2 \ldots n$ do

$$
\begin{aligned}
R_j &= {}_jG^Tv_i + \gamma_j \\
\ddot{\theta}_j &= D_jw_j - F_j^TR_j \\
v_j &= R_j + H_j\ddot{\theta}_j
\end{aligned}
$$

Figure 7. Computational scheme for recursive method [15]

To implement the computational scheme of Fig. 7, a reference point and a coordinate system have to be specified. This example follows the suggestion of BRANDL/JOHANNI/OTTER [4]. The local coordinate system $K(O_j, x_j, y_j, z_j)$ as introduced by DENAVIT/HARTENBERG [5] is used to express all magnitudes of body "j" The Origin O_j is used as reference point, see Fig. 8.

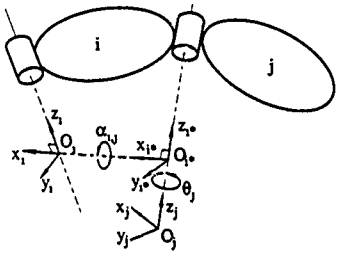


Figure 8: Denavit-Hartenberg coordinate systems

To express a vector or matrix in coordinates of a coordinate system $K(O_i, x_i, y_i, z_i)$, they will be denoted with upper left index, e g. ${}^i v = ({}^i v^T, {}^i \omega^T)^T$. The transformation matrix ${}^i K_j$ for sixdimensional velocities and forces between coordinate system $K(O_j, x_j, y_j, z_j)$ and $K(O_i, x_i, y_i, z_i)$ consists therefore of the known 3×3 orthonormal transformation matrix ${}^i T_j$,

$$ {}^i v = {}^i K_j \, {}^j v \quad ; \quad {}^i K_j = \begin{pmatrix} {}^i T_j & 0 \\ 0 & {}^i T_j \end{pmatrix} \quad . $$

The most critical part with respect to efficiency of the formalism is to transform coordinate system and reference point of the mass matrix M_j^P and the force g_j^P from $K(O_j, x_j, y_j, z_j)$ to the previous body $K(O_i, x_i, y_i, z_i)$.

$$ \begin{aligned} {}^i M_i &= {}^i M_i + {}^i_j G^T \{ {}_j K^j \, {}^j M_j^P \, {}_j K^{jT} \} \, {}^i_j G \quad , \\ {}^i g_i &= {}^i g_i - {}^i_j G^T \{ {}_j K^j \, {}^j g_j^P \} \quad . \end{aligned} $$

An efficient procedure is to do it step by step across the intermediate reference frame $K(O_{i*}, x_{i*}, y_{i*}, z_{i*})$, since then a maximum of zero- and one-operations are exploited. It can also be shown that it is more efficient to first transform the coordinates and then the reference points, i e.

$$ \begin{aligned} {}^i M_i &= {}^i M_i + {}^i_{i*} G^T \{ {}_{i*} K^{i*} \{ {}^{i*}_j G^{i*T} \{ {}_{i*} K^j \, {}^j M_j^P \, {}_{i*} K^{jT} \} \, {}^{i*}_j G^{i*} \} \, {}_{i*} K^{i*T} \} \, {}^i_{i*} G \quad , \\ {}^i g_i &= {}^i g_i - {}^i_{i*} G^T \{ {}_{i*} K^{i*} \{ {}^{i*}_j G^{i*T} \{ {}_{i*} K^j \, {}^j g_j^P \} \} \end{aligned} $$

Most parts of the formalism can be programmed directly in six-dimensional vector-notation, because the previously described scalar environment will take care of zero- and one-operations. However, for the previously described transformation operations, it is more efficient to decompose the six-dimensional operations into three-dimensional operations and write special, optimized methods for the three-dimensional coordinate transformations, where the different reference frames are just rotated about one coordinate axes. One example is the coordinate transformation between $K(O_j, x_j, y_j, z_j)$ and $K(O_{i*}, x_{i*}, y_{i*}, z_{i*})$ for a 3×3 tensor $\boldsymbol{A}$.

$$
\begin{aligned}
{}^{i*}\boldsymbol{A} &= {}^{i*}\boldsymbol{T}_j \quad {}^{j}\boldsymbol{A} \quad {}^{i*}\boldsymbol{T}_j{}^{T} \\
&= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} a_{22} + \cos\theta\, h_1 & a_{12} + \sin\theta\, h_1 & a_{13}\cos\theta - a_{23}\sin\theta \\ a_{21} + \sin\theta\, h_1 & a_{11} - \cos\theta\, h_1 & a_{13}\sin\theta + a_{23}\cos\theta \\ a_{31}\cos\theta - a_{32}\sin\theta & a_{31}\sin\theta + a_{32}\cos\theta & a_{33} \end{pmatrix}
\end{aligned}
$$

With $h_1 = (a_{11} - a_{22})\cos\theta - (a_{12} + a_{21})\sin\theta$, $h_2 = \cos\theta h_1$, $h_3 = \sin\theta h_1$, the transformation can be achieved with 11+ and 12*, besides 2 trigonometric function evaluations. For a symmetric matrix $\boldsymbol{A}$, this reduces to 8+ and 8*.

To avoid the transformation of the graviational forces from coordinates of the global reference frame into coordinates of the local reference frames of all bodies an important optimization of the formalism is to fictively accelerate the first body instead of adding gravitational forces [4], or, a little more efficient, to fictively accelerate the second body and add gravitational force to the first one [15]. Of course, if there are any springs or dampers between bodies, this will be useless. Additional minor optimizations can be achieved by moving the global reference frame to $K(O_1, x_1, y_1, z_1)$ at $t = 0$ and to define $K(O_n, x_n, y_n, z_n)$ in a way that $O_n = O_{(n-1)*}$ and the vector from O_n to the center of mass of body n becomes $\rho_n = (*, 0, *)^T$. Combining the previously described optimizations with the self-optimizing scalar environment, table 1 shows the minimum number of operations (without $2 * n$ trigonometric functions). For $n > 2$, the expected $O(n)$ behaviour

n	*	/	+	-	Σ
1	8	1	2	1	12
2	106	2	46	22	176
3	309	8	163	84	564
4	519	14	296	146	975
5	729	20	429	208	1386

Table 1: Minimum number of floating point operations

of the recursive formalism can be found. The minimum number of operations for $n > 2$ is $411n - 669$ which is faster than the number $470n - 420$ reported in [4], slightly faster than the number $432n - 688$ in [16], and faster for large systems than the number $428n - 737$ from [15].

4. Conclusion

The last years a large number of multibody dynamics formalisms have been published and some attempts have been made to compare them with respect to efficiency. These comparisons have shown that there is some potential to make a program faster by choosing the optimal formalism. The comparisons also have shown that the maximum efficiency of a formalism depends on the multibody system it is applied to. However, only some formalisms were compared on some benchmark problems. Furthermore, the compared formalisms have been strongly optimized under assumptions, which are not always applicable to more practical multibody systems. A better way to choose an efficient multibody dynamics formalism would be to apply a larger number of formalisms to a multibody system, compare them and select the optimal one. This paper describes an object-oriented programming environment in C++, which allows comfortable programming and modification of formalisms. Thus, a larger number of formalisms can be applied to a given system in reasonable time. To compare formalisms, they are automatically optimized and the floating point operations are counted. To verify a particular formalism, an executable program is generated, which runs with the minimum number of floating point operations.

5. References

[1] ARMSTRONG, W.W.: Recursive Solution to the Equations of Motion of an N-Link Manipulator, Proc. of the fifth World Congress on Theory of Machines and Mechanisms, pp. 1343-1346, Montreal, 1979.

[2] BAE, D.-S. AND HAUG, E.J.: A recursive formulation for constrained mechanical system dynamics: Part I. Open Loop Systems, Mech. Struct. & Mach., 15(3) 359-382, 1987.

[3] BOOCH, G.: Object oriented design with applications, The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1991.

[4] BRANDL, H.; JOHANNI, R.; OTTER, M.: A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix, Proc. of the IFAC/IFIP/IMACS Intern. Symposium of Theory of Robots, Vienna, Dec 3-5, 1986.

[5] DENAVIT, J.; HARTENBERG, R.S.: A kinematic notation for lower-pair mechanisms based on matrices, Journal of Applied Mechanics, 23, pp. 215-221, 1955.

[6] ELLIS, M. A.; STROUSTRUP, B.: The Annotated C++ Reference Manual, AT&T Bell Laboratories, New York, 1990.

[7] FEATHERSTONE, R.: Robot Dynamics Algorithms. Kluwer Academic Publishers, Boston, 1987.

[8] GRIES, D.: Compiler Construction for Digital Computers. John Wiley & Sons, Inc., New York, 1971.

[9] KECSKEMÉTHY, A., HILLER, M.. Object-oriented approach for an effective formulation of multibody dynamics. Second U S. National Congress on Computational Mechanics, Washington D.C., 1993.

[10] KECSKEMÉTHY. A.: Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen. PhD thesis Universität-GH-Duisburg, 1993.

[11] LEISTER, G.: Beschreibung und Simulation von Mehrkörpersystemen mit geschlossenen kinematischen Schleifen. VDI Fortschrittberichte Reihe 11, Nr. 167, VDI-Verlag, Düsseldorf, 1992.

[12] NIKRAVESH, P.E. AND GIM, G.: Systematic Construction of the Equations of Motion for Multibody Systems containing closed Kinematic Loops, ASME Design Automation Conference, 1989.

[13] REIN, U.: Implementation of a Recursive Projection Algorithm in a General Purpose Computer Program, Diplomarbeit am Institut A für Mechanik, Universität Stuttgart, 1989.

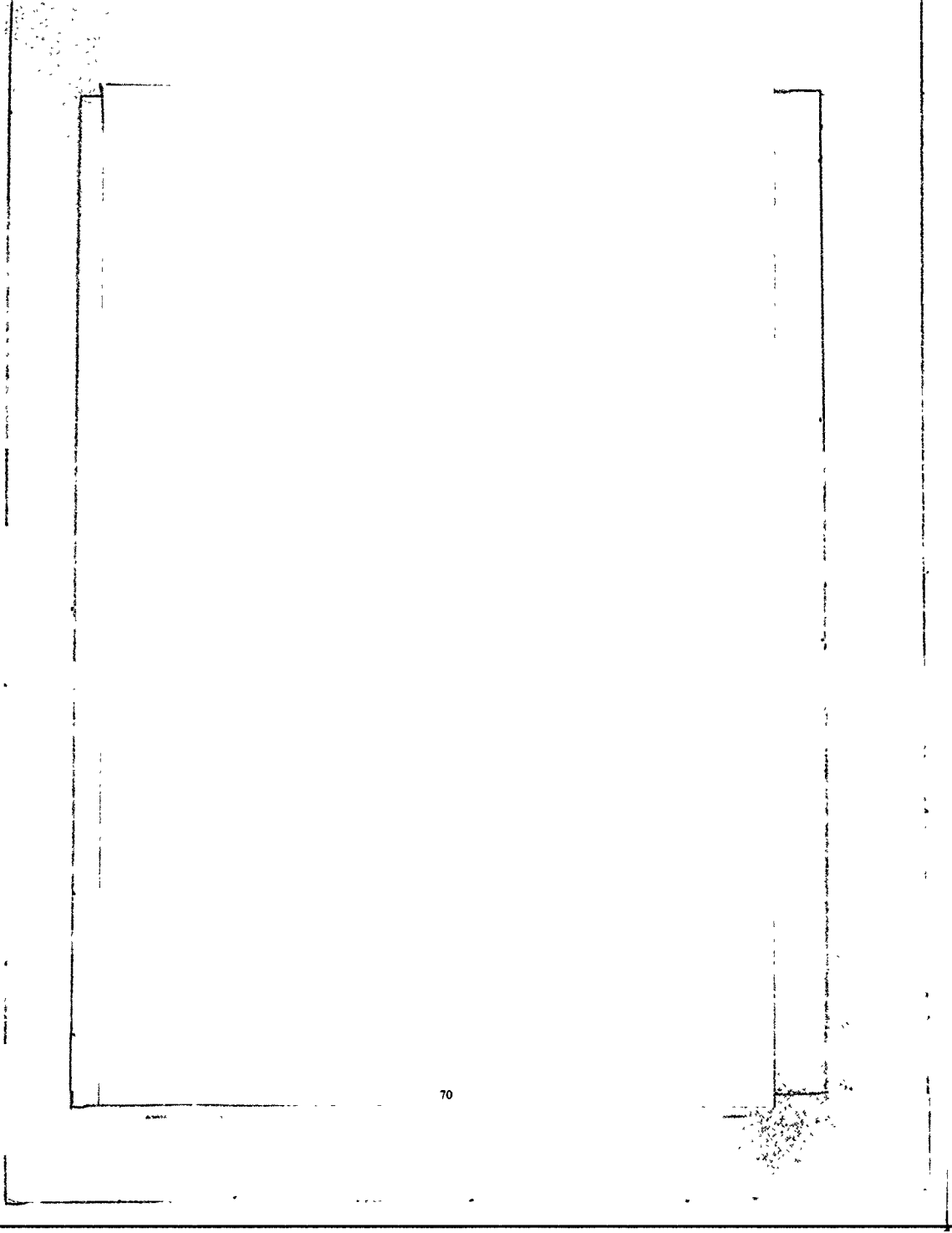
[14] SAYERS, M.W.: Symbolic Computer Language for Multibody Systems, J. Guidance, Vol. 14, No. 6, 1991.

[15] STELZLE, W.: Ein Vergleich rekursiver Verfahren zur Untersuchung der Dynamik baumstrukturierter Mehrkörpersysteme, ZAMM, 73 (1993) 4.

[16] VALÁŠEK, M.; LEISTER G.: Efficient implementation of multibody formalisms, Eighth World Congress on the Theory of Machines and Mechanisms, Prag, Czechoslovakia, August 26-31, 1991.

[17] WEHAGE, R.A.: Application of matrix partitioning and recursive projection to order n solution of constrained equations of motion. Proceedings of the 20th Biennial ASME Mechanisms Conference, Orlando, Florida, 1988.

[18] VERESCHAGIN, A.F.. Computer Simulations of the Dynamics of Complicated Mechanisms of Robot-Manipulators, Engineering Cybernetics, 6 65 70, 1974.



SPARSE-MATRIX GENERATION OF JACOBIANS FOR THE OBJECT-ORIENTED MODELLING OF MULTIBODY DYNAMICS

A. KECSKEMETHY
Fachgebiet Mechatronik
Universität Duisburg
Lotharstraße 1
W-4100 Duisburg 1
e-mail: kecs@mechatronik.uni-duisburg.de

ABSTRACT. Discussed in this paper is a new method for the generation of Jacobian matrices which is particularly designed for use in object-oriented modellings of multibody dynamics based on the transmission-element approach. The method starts from a description of multibody kinematics as a series of general mappings between manifolds, from which the overall Jacobian results — via the chain rule — as a sequence of matrix products. For these matrices, a new sparse-matrix scheme is suggested. Their "elements" are, besides zeroes, the well-known spatial transformation matrices and the local Jacobians of the individual transmission elements. It is shown how the main approaches for calculation of Jacobians in robotics can be viewed as particular decompositions and multiplication schemes of the sparse-matrices discussed above. Furthermore, two new schemes are derived which may be advantageous for dynamics calculations. A comparison of computation time for the Jacobian-based generation of dynamical equations and other methods (composite rigid body method, recursive methods) is carried out with a C++-implementation of a particular object-oriented modelling of multibody dynamics.

1. Introduction

The modelling of multibody dynamics has several applications in industrial engineering, which alltogether aim at reducing costs in development and production. A topic of particular interest is the on-line simulation of the kinematic and dynamic behaviour of mechanical systems: by means of this, the designer can obtain a better understanding of the influence of relevant parameters in the system, as well as precise quantitative information about effects which are difficult to estimate due to inherent nonlinearities of the system. For the simulation of a mechanical system, first the underlying mathematical equations must be stated, a task which, due to its complexity, can be carried out for even modestly sized systems only with the help of computers and specialized multibody programs. After two decades of research in this field, the issues which have prevailed all turn around the topic of *speed*. Currently, three different facets of this notion play a fundamental role in multibody dynamics: (1) *speed of modelling*, by virtue of which the design process can be better incorporated into the very short cycles of the modern production process, (2) *run-time speed*, which allows to use the models for animation, as hardware-in-the-loop components or as black-boxes in complex computation schemes as e.g. optimization and parameter estimation, and (3) *speed of broadening and re-design programming processes*, which is the main factor in applications in which the underlying equations constantly have to be extended and methodologies from different disciplines have to be combined to cope with the given problems, as it is the case e.g. in the field of *mechatronics*. In the beginnings of multibody

research, the emphasis was put mainly on the first of these aspects (Andrews and Kesavan 1975, Chace and Smith 1971, Orlandea et al. 1979, Wittenburg 1977, Sheth and Uicker Jr. 1972), while later in the eighties the rising complexity of applications forced the researchers to develop more effective formalisms. These arose first in the field of robotics (Featherstone 1987, Walker and Orin 1982) and carried over to general-purpose programs (Brandl et al. 1986, García de Jalón et al. 1986). The third issue is a topic of growing importance, because of the increasing necessity to amalgamate methods from different branches of science in order to be able to solve modern problems adequately. However, surprisingly almost no attention was paid to this issue until very recently.

The present *object-oriented modelling* of multibody systems arose from the demand for efficient, "hand-tailored" programs which are suitable for a robust and fast simulation of the dynamics of complex mechanical systems, as for example vehicle dynamics. In the approach introduced in Kecskeméthy and Hiller 1993, the multibody system is regarded as a collection of independent "intelligent" objects which take care of specific tasks autonomously. This allows to design problem-specific programs basically in the same way as preparing the input file for traditional self-contained programs. The objects are executable and can thus be incorporated very easily in superposed tasks, a property which makes them ideally suited for multidisciplinary projects. In this paper, the intuitive approach is extended to cover also efficient implementation techniques for Jacobian generation and composition. The structure-dependent information is described by a special form of sparse-matrix modelling, and well-known methods from robotics are incorporated into the general procedure. Also, some new computational schemes are derived which offer new possibilities for efficiency optimization in applications involving dynamics. The rest of the paper is organized as follows: In Section 2, the basic ideas of the present object-oriented modelling will be presented. Section 3 contains the particular modelling of Jacobians mentioned above, and Sections 4 and 5 encompass the main results and the conclusion of the present investigation, respectively.

2. Basic ideas of the proposed object-oriented multibody modelling

Object-oriented programming lends itself for combining efficient solution techniques with intuitive interfaces because of its ability to hide complex data and operation schemes behind plain terms (Booch 1991), just about the same way carefully introduced notions help to understand — and implement — complex relationships in mathematics. The key issue in object-oriented programming is not the implementation part — which is feasible under any of the current programming languages — but the *design* of an appropriate decomposition of the physical system into physical objects whose behaviour can be mapped as directly as possible to corresponding executable programming modules. A crucial issue in the design process is whether the objects reflect more the details of the particular implementation at hand than the inherent physical properties of the system being modelled. This distinguishing mark is most appropriately characterized by the notions of "data-driven" and "responsibility-driven" approaches.

The traditional practice in program design is the data-driven approach. Here, the objects are defined as encompassing units of the data structures needed by the algorithms. This can facilitate the implementation of a particular methodology substantially. However, because of the dependency of object definition on its implementation, the objects are not

only dependent on its intrinsic behaviour, but also on issues such as programming style, compiler features, etc. and thus for many people the advantages of object-oriented programming do not become apparent. In contrast to this, responsibility-driven approaches view a program as a collection of either active or passive entities which are endowed with enough "intelligence" to carry out tasks — their so-called "*responsibilities*" — rather autonomously. A particular model in this setting is the "*client/server model*" of Wirfs-Brock and Wilkerson 1989: the tasks in a running program reflect here typical transactions of everyday life, where a server performs actions requested by a client according to a specific contract, the client not having to be informed about more details of the procedure or about the tools or resources used therein than needed for giving out the assignment. The advantage of this approach is that, after carrying out the modelling, an intuitive programming environment results which corresponds in a high degree to conventional graphic user interfaces used in CAD environments for the rapid prototyping of multibody systems. Thus, there is almost no penalty for using programming tools rather then graphic user interfaces, but a high gain in modularity and further applicability to further tasks. The crux of this approach is to find a particular view of the problem which indeed facilitates the dissection of a physical system into neatly defined, implementation-independent "clients" and "servers". This section gives a short overview of a particular method of achieving this representation in order to make the subsequent expositions clearer. The interested reader is refered to the more elaborate expositions of this subject in Kecskeméthy 1993 and Kecskeméthy and Hiller 1993.

2.1. OBJECTS IN MULTIBODY SYSTEMS

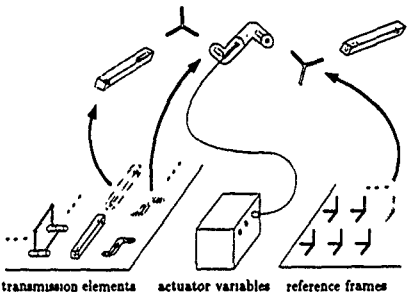


Figure 1: Objects in multibody systems

The key idea of the proposed modelling is to regard the multibody system as a collection of *kinetostatic transmission elements* which map motion-, force- and mass-related information from a set of *state objects* to another. State objects are able to hold and return information about position, velocity, acceleration and load at an arbitrary location. Conceptually, the multibody system is obtained by attaching or "plugging in" specific copies of state objects

to the in- and outputs of the transmission elements (Fig. 1).

The state objects are subdivided into scalar variables and spatial objects. The "spatial" state object embraces information about the motion of a reference frame $\mathcal{K}$ with respect to an implicitly defined reference frame $\mathcal{K}_0$, as well as about the load being applied to the origin of $\mathcal{K}$. The motion information is comprised by the orthogonal rotation matrix R transforming vector components from $\mathcal{K}$ to $\mathcal{K}_0$, the radius vector r from the origin of $\mathcal{K}_0$ to the origin of $\mathcal{K}$, the angular and linear velocity ω and v, respectively, and the corresponding acceleration terms $\dot{\omega}$ and a. Besides this, also the force f and the torque τ being applied to the frame are stored. Free vectors v, a and τ are assumed to be measured at the origin of $\mathcal{K}$, and all vectors to be decomposed in $\mathcal{K}$. For notational convenience, angular and linear components of velocity and load are collected in twist and wrench vectors (Hunt 1990) $\underline{t} = [\omega^T, v^T]^T$ and $\underline{w} = [\tau^T, f^T]^T$, respectively.

Scalar state variables β serve as actuator inputs for joints or as outputs for sensors performing scalar measurements. They also integrate information about position, velocity and acceleration as well as a load component describing the generalized force related to the variable. The sign of the latter is defined such that under a virtual displacement $\delta\beta$ a positive load Q_β *supplies* energy to the system. By abuse of notation, we will refer to both the value of the variable and to the complete set of its kinetostatical information (i.e. position, velocity, acceleration and force) with the same symbol β. However, under computational aspects it is wise to explicitly discern linear and angular variables whose values lie on the real line $\mathbf{R}$ and the circle S^1, respectively. This differentiation is observed in some cases below.

A basic kinetostatic transmission is capable of carrying out *motion transmission* and *force transmission* (Fig. 2).

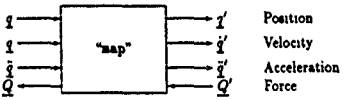


Figure 2: A simple transmission element

The operation of *motion transmission* consists of the three sub-operations

$$\left.\begin{array}{lll} \underline{\text{position:}} & q' = \phi(q) & \\ \underline{\text{velocity:}} & \dot{q}' = J_\phi \dot{q}\,; \; J_\phi = \dfrac{\partial \phi}{\partial q} & \\ \underline{\text{acceleration:}} & \ddot{q}' = J_\phi \ddot{q} + \dot{J}_\phi \dot{q} & \end{array}\right\}, \qquad (1)$$

where $q \in \mathbf{R}^n$, $q' \in \mathbf{R}^m$, and $J_\phi \in \mathbf{R}^{m \times n}$ is the *Jacobian* of the transmission element.

Under the assumption of an *ideal* transmission element, i.e. one that neither generates nor consumes power, a force-transmission mapping can be defined from the equality of virtual work at the input and output of the transmission element,

$$\delta q^T Q = \delta q'^T Q' \; .$$

After subsituting $\delta \underline{q}' = J_\phi \delta \underline{q}$ and noting that this condition must hold for all $\delta \underline{q} \in \mathbb{R}^n$, one obtains the *force transmission* function

$$\underline{\text{force:}} \qquad \underline{Q} = J_\phi^T \underline{Q}' \ . \tag{2}$$

Note that this force mapping is directed in *opposite* direction to the motion transmission. This property is quite general and can be verified in many applications (e.g. Luh and Zheng 1985).

2.2. GEOMETRIC MODELLING OF TRANSMISSION ELEMENTS

A general transmission element can be modelled mathematically as a mapping from a general "input" manifold X to an "output" manifold X' (Fig. 3).

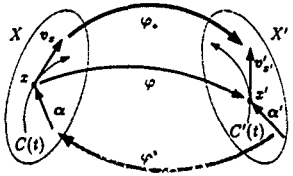


Figure 3: Mappings between two manifolds

The model of a manifold is used here because it gives a means of considering combinations of spatial and scalar state objects as single geometric entities

$$X = \underbrace{\mathcal{K} \times \cdots \times \mathcal{K}}_{n_\mathcal{K} \text{ copies}} \times \underbrace{\mathbf{R} \times \cdots \times \mathbf{R}}_{n_\mathbf{R} \text{ copies}} \times \underbrace{S^1 \times \cdots \times S^1}_{n_{S^1} \text{ copies}} \ , \tag{3}$$

where $n_\mathcal{K}$ is the number of reference frames, and $n_\mathbf{R}$ and n_{S^1} are the number of linear and rotational variables, respectively. Given a mapping φ: $X \to X'$, $x \in X \mapsto x' = \varphi(x) \in X'$, the *differential map* or *push forward* function φ_* maps *tangent vectors* v_x in the tangent space T_x onto tangent vectors $v'_{x'}$ in the tangent space $T'_{x'}$. This mapping is formally defined by (Choquet-Bruhat and DeWitt-Morette 1989)

$$\varphi_*: v_x = \frac{dC(t)}{dt} \ \mapsto \ v'_{x'} = \varphi_*(v_x) \doteq \frac{d(\varphi(C(t)))}{dt} \ , \tag{4}$$

where $C(t)$ is a general trajectory on X and $C'(t)$ is its image under φ.

In differential geometry, an additional geometric object is defined which is the linear mapping of tangent vectors to the set of real numbers. Such objects α', denominated *cotangent vectors*, span a cotangent vector space $T'^*_{x'}$ of same dimension as T_x. Associated with φ, there exists a function φ^* which maps cotangent vectors α' at the output of φ to corresponding vectors α at its input. This mapping, denominated the *pull-back* function, is defined by

$$\varphi^*: T'^*_{x'} \to T^*_x \ , \ \alpha' \mapsto \alpha = \varphi^*(\alpha') \ : \ \alpha(v_x) \equiv \alpha'(v'_{x'}) \ . \tag{5}$$

By regarding the operation of computation of virtual power as a linear mapping of tangent vectors to real numbers, it is easy to verify that cotangent vectors correspond to forces and that the pull-back function models the force transmission (Kecskeméthy 1993).

2.3. CONCATENATION OF TRANSMISSION ELEMENTS

By the manifold analogy, chains of interconnected transmission elements can be modelled as composite functions. From this modelling, the laws for Jacobian generation simply result as a consequence of applying the chain rule. For example, for the sequence of two mappings f and g in Fig. 4 the following relationships are obtained (von Westenholz 1981):

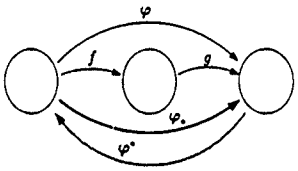


Figure 4: Composition of mappings

$$\left.\begin{array}{lrcll} \underline{\text{position:}} & \varphi & = & g \circ f & \text{(forwards)} \\ \underline{\text{tangent vectors:}} & \varphi_\circ & = & (g \circ f)_\circ = g_\circ \circ f_\circ & \text{(forwards)} \\ \underline{\text{cotangent vectors:}} & \varphi^* & = & (g \circ f)^* = f^* \circ g^* & \text{(backwards)} \end{array}\right\}. \qquad (6)$$

Note that, in general, the operators in Eq. (6) are applied from *right to left*, as in $(g \circ f)(x) \equiv g(f(x))$. However, it is common practice in kinematics to view sequences of transmission elements as in Fig. 4 as concatenations starting at the inertial frame and moving from left to right as one proceeds to the tips of the chains. Thus, in the functional notation the ordering of transmission elements is *reverse* to the ordering of the "kinematical" notation. Also, it is interesting to note that for the force transmission the elements are again traversed in the reverse order to the motion transmission sequence.

A general composite transmission element is modelled as a chain of simpler transmission elements which can be elementary as e.g. links, joints etc., but also more elaborate subsystems such as closed loops, wheel suspensions or complete control mechanisms. This has no effect on the definition of the basic mappings, which are just a concatenation of the corresponding mappings of the components. However, for computational efficiency it is necessary that the *structure* of the interconnection structure is also taken into account in order to avoid redundant operations in the calculation of products such as in Eq. (6). This is accomplished by the method discussed in the sequel.

3. Modelling of Jacobians

The concept of the transmission element introduced in the previous section gives a means of unifying the different approaches for Jacobian evaluation used in multibody dynamics. In

this setting, Jacobians are regarded as the differential maps of the underlying transmission functions, i.e. functions transporting tangent vectors at the input to tangent vectors at the output of the transmission elements. The basic idea of the approach is that, given a chain of transmission elements as shown in Fig. 5, and defined by the composite function

$$\varphi = \varphi_n \circ \cdots \circ \varphi_1 = \varphi_n\big(\cdots \varphi_1(\cdot)\big) \ , \tag{7}$$

the corresponding differential mapping is obtained, according to the chain rule, as

$$\varphi_* = \varphi_{n*} \circ \cdots \circ \varphi_{1*} \ , \tag{8}$$

where the φ_{i*} are the differential maps of the individual transmission elements.

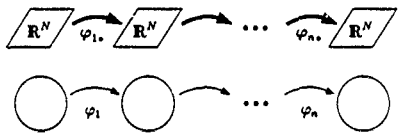


Figure 5: Concatenation of transmission elements

For simplification of the concatenation operation, it is advantageous to merge all input and output manifolds of the individual transmission elements into one *global* manifold and to regard the individual transmission elements as mappings within this space. Then, the tangent spaces of the inputs and outputs of the transmission elements are all identical, and the overall Jacobian results from a composition of several linear maps $\mathbf{R}^N \to \mathbf{R}^N$, where N is the dimension of the resulting global manifold (Fig. 5). In the following, a method for the efficient computation of this kind of composition is discussed.

3.1. LOCAL STRUCTURE OF JACOBIANS

For the analysis of the structure of Jacobians, consider a typical mechanical part of a general multibody system — here denominated a "*mechanical component*" C_i — consisting of a series of movable rigid or deformable bodies mounted on a floating base (Fig. 6). This object has as inputs the motion of the reference frame $\mathcal{K}_i$ of the floating base, as well as a set of f_{c_i} internal variables $\underline{\beta}_{(i)}$ representing the (independent) generalized coordinates (e.g. joint variables or modal coordinates for elastic deformation). The outputs are conformed by frames attached at places of interest or further couplings to the moving bodies.

Regarding this object as a general mapping, the differential mapping transports two "types" of tangent vectors: those related to rigid-body motion — i.e. the "*twists*" — and those related to the generalized coordinates — i.e. n-tuples in a linear subspace $V_\beta \subset \mathbf{R}^{f_{c_i}}$. Because of the linearity of differential maps, these two contributions can be treated separately and added together subsequentially.

First, consider the tangent vectors related to rigid-body motion. For these quantities, four specifications are necessary:

(1) the reference frame $\mathcal{K}_j$ whose motion is measured,

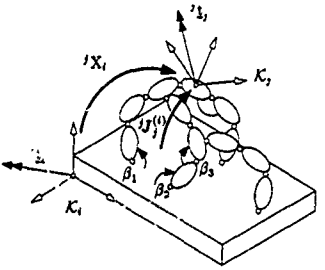


Figure 6: Model of a mechanical component

(2) the reference frame $\mathcal{K}_i$ with respect to which motion is measured,

(3) the reference frame $\mathcal{K}_\ell$ (fixed with respect to $\mathcal{K}_j$) whose origin O_ℓ is taken as reference point for the linear velocity of $\mathcal{K}_j$, and

(4) the reference frame $\mathcal{K}_k$ in which the vectors are decomposed.

These four indices are arranged in the general case as $^{k}_{i}\underline{t}_{j(\ell)}$. However, for most applications, the number of indices can be reduced, because (a) the reference frame $\mathcal{K}_i$ is mostly the inertial frame, and (b) the reference point for the velocity is frequently identical to the origin of the reference frame whose motion is measured. Thus, a reduced notation involving only the indices j and k as introduced above is used in the sequel.

The transmission of twists involves a linear transformation of the twist $^{i}\underline{t}_{i}$ at the input of the mechanical component to a twist $^{j}\underline{t}_{j}$ at its output. Here, equal indices for the frame of decomposition and the current moving frame are introduced for further simplification of notation. This general transformation can be written compactly as

$$^{j}\underline{t}_{j} = {}^{j}X_{i}\,{}^{i}\underline{t}_{i}\,, \tag{9}$$

where $^{j}X_{i}$ is the *"spatial transformation matrix"*

$$^{j}X_{i} = \begin{bmatrix} {}^{i}R_{j}^{T} & 0 \\ -{}^{i}_{j}\tilde{r}_{j}\,{}^{i}R_{j}^{T} & {}^{i}R_{j}^{T} \end{bmatrix}\,. \tag{10}$$

Here, $^{i}R_{j}$ is the 3×3 othogonal rotation matrix transforming vector components from $\mathcal{K}_j$ to $\mathcal{K}_i$, $^{k}_{j}r_{j}$ is the radius vector from origin O_i to origin O_j decomposed in frame $\mathcal{K}_k$ (in this particular case with $k = j$) and the tilde is the well-known operator for constructing a skew-symmetric matrix out of a vector such that $\tilde{r}\,x \equiv r \times x$.

Concatenation of spatial transformations is achieved by multiplication of their respective matrix representations, viz.

$$ {}^{k}X_{i} = {}^{k}X_{j}\,{}^{j}X_{i} = \begin{bmatrix} {}^{i}R_{i}^{T} & 0 \\ -{}^{k}\tilde{r}_{ik}\,{}^{i}R_{i}^{T} & {}^{i}R_{i}^{T} \end{bmatrix} , \tag{11} $$

where

$$ {}^{i}R_{i}^{T} = {}^{j}R_{i}^{T}\,{}^{i}R_{j}^{T} , \tag{12} $$

$$ {}^{k}r_{ik} = {}^{k}r_{jk} + {}^{j}R_{k}^{T}\,{}^{j}r_{ij} . \tag{13} $$

Also, it is possible to decompose the general spatial transformation matrix in a part ${}^{j}X_{i}^{P}$ involving pure translation (representing the change of reference point for the velocity) and a part ${}^{j}X_{i}^{R}$ involving pure rotation (representing the change of coordinates)

$$ {}^{j}X_{i} = {}^{j}X_{i}^{P}\,{}^{j}X_{i}^{R} \; ; \; {}^{j}X_{i}^{P} = \begin{bmatrix} I_{3} & 0 \\ -{}^{j}\tilde{r}_{ij} & I_{3} \end{bmatrix} , \; {}^{j}X_{i}^{R} = \begin{bmatrix} {}^{i}R_{j}^{T} & 0 \\ 0 & {}^{i}R_{j}^{T} \end{bmatrix} . \tag{14} $$

Note that here the radius vector ${}^{j}r_{ij}$ and thus ${}^{j}X_{i}^{P}$ itself is decomposed in $\mathcal{K}_{j}$. However, the effect of a pure translation can also be related to any other frame $\mathcal{K}_{\ell}$ by means of

$$ {}^{(j)}X_{i}^{P} = {}^{\ell}X_{j}^{R}\,{}^{j}X_{i}^{P}\,{}^{\ell}X_{j}^{R^{-1}} . \tag{15} $$

In this notation, i marks the moving frame regarded, ℓ the frame of decomposition, and j the origin taken as the reference point for the velocity of $\mathcal{K}_{i}$. For example, changing the reference point from the origin O_{i} to the origin O_{j} within the same rigid body, while mantaining the frame of decomposition as $\mathcal{K}_{i}$, is achieved via the transformation

$$ {}^{i}\underline{t}_{j} = {}^{i(j)}X_{i}^{P}\,{}^{i}\underline{t}_{i} . \tag{16} $$

Next, the transmission of the time-derivatives of the generalized coordinates $\dot{\underline{\beta}}_{(i)}$ to the twist of an arbitrary output frame $\mathcal{K}_{j}$, decomposed in a frame $\mathcal{K}_{k}$, is considered. The corresponding linear mapping is denoted as

$$ {}^{k}\underline{t}_{j} = {}^{k}J_{j}^{(i)}\,\dot{\underline{\beta}}_{(i)} , \tag{17} $$

where ${}^{k}J_{j}^{(i)}$ represents the *local Jacobian* of the transmission element. Here, the indices have the following meaning: i records which set of generalized coordinates $\underline{\beta}_{(i)}$ is being used, j the frame $\mathcal{K}_{j}$ whose motion is measured, and k the frame $\mathcal{K}_{k}$ used for decomposition. For *elementary* transmission elements, the Jacobians are usually given *a priori*. In this case, the frame of decomposition is identical to the output frame, and the Jacobian has mostly constant coefficients. For example, a revolute joint turning $\mathcal{K}_{i}$ to $\mathcal{K}_{j}$ about $z_{i} \equiv z_{j}$ (regarded as component C_{i}) has a local Jacobian of ${}^{j}J_{j}^{(i)} = [0,0,1,0,0,0]^{T}$, while a similar prismatic joint generates the local Jacobian ${}^{j}J_{j}^{(i)} = [0,0,0,0,0,1]^{T}$. For *composite* transmission elements, the initially given local Jacobians are gradually "moved" up or down the branches of the transmittion elements to calculate the effects of the variation of local

generalized coordinates in other places of the multibody system. Specifically, changing both origin and frame of decomposition from $\mathcal{K}_j$ to $\mathcal{K}_k$ is achieved with the transformation

$$ {}^{k}J_{k}^{(i)} = {}^{k}X_{j}\, {}^{j}J_{j}^{(i)} \quad . \tag{18} $$

Changing only the frame of decomposition from $\mathcal{K}_j$ to $\mathcal{K}_k$, or changing only the point of reference from $\mathcal{O}_j$ to $\mathcal{O}_\ell$, result, respectively, in

$$ {}^{k}J_{j}^{(i)} = {}^{k}X_{j}^{R}\, {}^{j}J_{j}^{(i)} \quad , \tag{19} $$

$$ {}^{j}J_{j(\ell)}^{(i)} = {}^{j(\ell)}X_{j}^{P}\, {}^{j}J_{j}^{(i)} \quad . \tag{20} $$

In this last equation, the general notation ${}^{j}J_{j(\ell)}^{(i)}$ includes, besides the indices i, j, k described above, an additional index ℓ characterizing the origin $\mathcal{O}_\ell$ used as point of reference for the velocity of $\mathcal{K}_j$.

Combination of twist and generalized coordinate transmission yields the overall local Jacobian. For example, from equations (9) and (18) the resulting local Jacobian for the output frame $\mathcal{K}_j$ in Fig. 6 is

$$ {}^{j}\underline{\mathfrak{t}}_{j} = \left[{}^{j}X_{i}\, ,\, {}^{j}J_{j}^{(i)} \right] \begin{bmatrix} {}^{i}\underline{\mathfrak{t}}_{i} \\ \underline{\dot{\beta}}_{(i)} \end{bmatrix} \quad . \tag{21} $$

3.2. CONCATENATION OF JACOBIANS

Local Jacobians in kinematical subsystems consisting of many closed loops are most efficiently evaluated by applying hand-tailored formulas (Hiller and Kecskeméthy 1989) or automated symbolical manipulation schemes (Kecskeméthy and Hiller 1992). However, for series-connections of mechanical components, an automated approach can be stated which leads to a near-minimum computational effort. This is discussed in the sequel for a sequence of serially connected transmission elements as in Fig. 7. The generalization to general tree-type structures of connected mechanical components is straight-forward and shall not be regarded here.

Assuming that the mechanical components C_i in the sequence are already modelled, the local mappings $\varphi_i^{\text{local}} : (\mathcal{K}_i, \underline{\beta}_{(i)}) \mapsto \mathcal{K}_{i+1}$ as well as the corresponding local differential mappings $\varphi_{(i)}^{\text{local}}{}_{*} : (\underline{\mathfrak{t}}_i, \underline{\dot{\beta}}_{(i)}) \mapsto \underline{\mathfrak{t}}_{i+1}$ are expected to be known. These local differential mappings are now embedded in the global tangent space as discussed above. In this global space, a tangent vector has the structure $v = [{}^{1}\underline{\mathfrak{t}}_{1}^{\mathrm{T}}, \ldots, {}^{n+1}\underline{\mathfrak{t}}_{n+1}^{\mathrm{T}}, \underline{\dot{\beta}}_{(1)}^{\mathrm{T}}, \ldots, \underline{\dot{\beta}}_{(n)}^{\mathrm{T}}]^{\mathrm{T}}$, and for the differential mapping the function

$$ \varphi_{*} : v \mapsto v' = \varphi_{*}\, v \tag{22} $$

implies a *sparse* "globalized" Jacobian φ_{*} consisting mostly of zeroes and ones. For example, the globalized Jacobian for the mapping φ_i in Fig. 7 has the structure:

$$
\begin{array}{c}
\begin{array}{cccccccccc} \mathcal{K}_1 & \cdots & \mathcal{K}_i & \mathcal{K}_j & \mathcal{K}_k & \cdots & \mathcal{K}_{n+1} & \underline{\beta}_{(1)} \cdots & \underline{\beta}_{(i)} & \underline{\beta}_{(j)} \; \cdots \; \underline{\beta}_{(n)} \end{array} \\
\varphi_{(i)*} = \left[\begin{array}{ccccc|cccc}
I_6 & & & & & & & & \\
 & \ddots & & & & & \vdots & & \\
 & & I_6 & & & & & & \\
\cdots & {}^{j}X_i & 0 & \cdots & & & {}^{j}J_j^{(i)} & & \cdots \\
 & & & I_6 & & & & & \\
 & \vdots & & & \ddots & & \vdots & & \\
 & & & & I_6 & & & & \\
\hline
 & & & & & I_{fc_1} & & & \\
 & & & & & & \ddots & & \\
 & & 0 & & & & I_{fc_i} & & \\
 & & & & & & & I_{fc_j} & \\
 & & & & & & & & \ddots \\
 & & & & & & & & I_{fc_n}
\end{array}\right]
\begin{array}{l} \mathcal{K}_1 \\ \vdots \\ \mathcal{K}_i \\ \mathcal{K}_j \\ \mathcal{K}_k \\ \vdots \\ \mathcal{K}_{n+1} \\ \underline{\beta}_{(1)} \\ \vdots \\ \underline{\beta}_{(i)} \\ \underline{\beta}_{(j)} \\ \vdots \\ \underline{\beta}_{(n)} \end{array}
\end{array} \quad . \qquad (23)
$$

The first $n+1$ block rows and columns consist of quadratic submatrices of order 6 representing spatial transformations, while the last n block rows and columns correspond to the generalized coordinates of the components, the submatrices I_{fc_i} being identity matrices of dimension f_{c_i}. It is interesting to realize that the structure of this matrix *mirrors* the three basic types of variables that arise in the general transmission element: (1) *input variables*, characterized by non-zero elements appearing in the correspondig column above the diagonal (in this example $\mathcal{K}_i$ and $\underline{\beta}_{(i)}$), (2) *output variables*, distinguished by non-zero entries arising in the corresponding row (in this example $\mathcal{K}_j$), and (3) *through variables*, featuring only zero elements beside the diagonal in the corresponding row and column.

With this structure, the Jacobian of a composite system is easily established by multiplication. For example, with the globalized Jacobian of the mapping φ_j in Fig. 7,

$$
\begin{array}{c}
\begin{array}{cccccccccc} \mathcal{K}_1 & \cdots & \mathcal{K}_i & \mathcal{K}_j & \mathcal{K}_k & \cdots & \mathcal{K}_{n+1} & \underline{\beta}_{(1)} \cdots & \underline{\beta}_{(i)} & \underline{\beta}_{(j)} \; \cdots \; \underline{\beta}_{(n)} \end{array} \\
\varphi_{(j)*} = \left[\begin{array}{ccccc|cccc}
I_6 & & & & & & & & \\
 & & & & & & & & \\
 & I_6 & & & & & & & \\
 & & I_6 & & & & & & \\
\cdots & & {}^{k}X_j & 0 & \cdots & & & {}^{k}J_k^{(j)} & \\
 & & \cdot & & \cdot & & & \cdot & \\
 & & & & I_6 & & & & \\
\hline
 & & & & & I_{fc_1} & & & \\
 & & & & & & \cdot & & \\
 & & 0 & & & & I_{fc_i} & & \\
 & & & & & & & I_{fc_j} & \\
 & & & & & & & & I_{fc_n}
\end{array}\right]
\begin{array}{l} \mathcal{K}_1 \\ \vdots \\ \mathcal{K}_i \\ \mathcal{K}_j \\ \mathcal{K}_k \\ \vdots \\ \mathcal{K}_{n+1} \\ \underline{\beta}_{(1)} \\ \vdots \\ \underline{\beta}_{(i)} \\ \underline{\beta}_{(j)} \\ \underline{\beta}_{(n)} \end{array}
\end{array} \quad , \qquad (24)
$$

the Jacobian of the composite function $\overline{\varphi} = \varphi_{(j)} \circ \varphi_{(i)}$ is obtained as the product $\overline{\varphi}_* =$

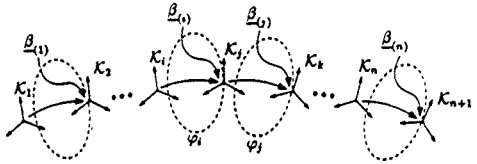


Figure 7: Concatenation of mechanical components

$\varphi_{(j)_o} \circ \varphi_{(i)_o}$, yielding

$$\overline{\varphi}_o = \left[\begin{array}{cccccc|cccc} I_6 & & & & & & & & & \\ & \ddots & \vdots & \vdots & & & & \ddots & \vdots & \\ & & I_6 & & & & & & & \\ & \cdots & {}^jX_i & 0 & \cdots & & \cdots & {}^j_j J^{(i)}_j & \cdots & \\ & \cdots & {}^kX_j\,{}^jX_i & 0 & 0 & \cdots & \cdots & {}^kX_j\,{}^j_j J^{(i)}_j & {}^kJ^{(j)}_k & \cdots \\ & & \vdots & & \ddots & & & \vdots & & \\ & & & & & I_6 & & & & \\ \hline & & & & & & I_{fc_1} & & & \\ & & & & & & & \ddots & & \\ & & & & & & & & I_{fc_i} & \\ & & & 0 & & & & & & I_{fc_j} \\ & & & & & & & & & & \ddots & \\ & & & & & & & & & & & I_{fc_n} \end{array}\right] \qquad (25)$$

$\mathcal{K}_1 \;\cdots\; \mathcal{K}_i \;\; \mathcal{K}_j \;\; \mathcal{K}_k \;\cdots\; \mathcal{K}_{n+1} \;\; \underline{\beta}_{(1)} \;\cdots\; \underline{\beta}_{(i)} \;\; \underline{\beta}_{(j)} \;\cdots\; \underline{\beta}_{(n)}$

$\mathcal{K}_1 \;\cdots\; \mathcal{K}_i \;\; \mathcal{K}_j \;\; \mathcal{K}_k \;\cdots\; \mathcal{K}_{n+1} \;\; \underline{\beta}_{(1)} \;\cdots\; \underline{\beta}_{(i)} \;\; \underline{\beta}_{(j)} \;\cdots\; \underline{\beta}_{(n)}$

After carrying out the substitutions

$$ {}^kX_i = {}^kX_j\,{}^jX_i \;, \qquad (26)$$

$$ {}^kJ^{(i)}_k = {}^kX_j\,{}^j_j J^{(i)}_j \;, \qquad (27)$$

one obtains again a globalized Jacobian of the same structure as those in equations (23) and (24). Note that in this derivation both the point of reference and the frame of decomposition are taken from the output frame. Alternatively, one may relate both the point of reference and the decomposition frame to the inertial frame, which is here assumed to be $\mathcal{K}_1$. Then new representations of the $\varphi_{(j)_o}$ are obtained via the transformations

$$ \varphi'_{(j)_o} = B\,\varphi_{(j)_o} B^{-1} \;, \qquad (28)$$

where

$$ B = \mathrm{diag}\left\{ I_6, {}^1X_2, \cdots, {}^1X_{n+1}, I_{fc_1}, \cdots, I_{fc_n} \right\} \qquad (29)$$

Another approach is to use as reference point for velocities the origin of the corresponding output frame, but as decomposition frame again the inertial frame. Such a representation

is obtained analogously to Eq. (28) by substituting in Eq. (29) the terms ${}^{1}X_j$ by ${}^{1}X_j^R$. Similarly, the representation using the origin of the inertial frame as reference point for the velocities and the output frame as frame of decomposition is obtained by substituting in Eq. (29) ${}^{j(1)}X_j^P$ for ${}^{1}X_j$. The relevance of these different schemes will become evident in Section 4. For the moment, it is of interest to appreciate how different methods of applying spatial transformations eventually lead to completely different schemes of Jacobian evaluation.

3.3. SPARSE MATRIX MODELLING OF JACOBIANS

In order to take advantage of the sparse structure of the globalized Jacobians in operations such as the multiplication discussed above, a particular scheme derived from well-known sparse-matrix techniques is introduced. To this end, the general differential mapping is decomposed into two parts:

$$\varphi_* = I_\varphi + \hat{\varphi}_* \quad . \tag{30}$$

The first part corresponds to the diagonal of φ_*

$$I_\varphi = \mathrm{diag}\left\{ \delta_{\mathrm{sel}}(\mathcal{K}_1)\, I_6 \, , \ldots \, , \delta_{\mathrm{sel}}(\mathcal{K}_{n+1})\, I_6 \, , I_{fc_1} \, , \ldots \, , I_{fc_n} \right\} \quad , \tag{31}$$

where σ_{sel} is a BOOLEan variable characterizing output variables,

$$\sigma_{\mathrm{sel}}(\mathcal{K}_i) = \begin{cases} 1 & \text{if } i\text{th reference frame is an output} \\ 0 & \text{otherwise} \end{cases} \tag{32}$$

and I_{fc_i} is an identity matrix of same dimension as the number of internal coordinates of component C_i. The remaining matrix $\hat{\varphi}_*$ now has very few elements, which can be stored using a modification of the well-known sparse row-wise format (Pissanetzky 1984). This scheme is as follows:

(1) In a field **elements**, store pointers to the non-vanishing entries of matrix $\hat{\varphi}_*$ (local Jacobians ${}^{i}J_i^{(i)}$ and spatial transformations ${}^{j}X_i$) row by row. The order of the elements within a row is immaterial, but rows must be stored in order of their appearence from top to bottom.

(2) A field **columns** contains the columns where the entries of **elements** are placed in $\hat{\varphi}_*$.

(3) A further field **rows** holds the row indices of $\hat{\varphi}_*$ where non-zero elements occur.

(4) Finally, the places in **elements** and **columns** where new rows (of $\hat{\varphi}_*$) start are recorded in a field **row-pointers**.

For example, the sparse-matrix storage scheme for the Jacobian $\overline{\varphi}_*$ resuling from Eq. (25) would look like

$$\begin{aligned} \textbf{elements} &= [{}^{j}X_i \, , {}^{j}J_j^{(i)} \, , {}^{k}X_i \, , {}^{k}J_k^{(i)} \, , {}^{k}J_k^{(j)}] \\ \textbf{columns} &= [i \, , n+i+1 \, , i \, , n+i+1 \, , n+i+2] \\ \textbf{rows} &= [j \, , k] \\ \textbf{row-pointers} &= [1 \, , 3] \end{aligned}$$

Thus, it takes only a few additional entries (14 elements, in this example), to store the structure of the Jacobian. It should be noted that this information is basically equivalent to graph-theoretic methods such as the adjacency matrix approach (Wittenburg 1977). The advantage of this representation is that it leads directly to highly optimized formulas without the need of deriving them from more general notions.

We now study the effect of the decomposition (30) on the multiplication of two differential maps. Multiplication of all terms gives rise to five basic operations which have to be carried out to obtain again the decomposition introduced in Eq. (30):

$$\varphi_{C_\bullet} = \varphi_{B_\bullet} \circ \varphi_{A_\bullet} = \underbrace{\underbrace{I_{\varphi_B} \circ I_{\varphi_A}}_{I_{\varphi_C}} + \underbrace{I_{\varphi_B} \circ \hat{\varphi}_{A_\bullet}}_{(3)\equiv\hat{\varphi}_{A_\bullet}}}_{(5)} + \underbrace{\underbrace{\hat{\varphi}_{B_\bullet} \circ I_{\varphi_A}}_{(2)} + \underbrace{\hat{\varphi}_{B_\bullet} \circ \hat{\varphi}_{A_\bullet}}_{(1)}}_{(4)} . \tag{33}$$

By further scrutinizing these terms, one finds the following properties:

(1) This is a general sparse matrix product. It consists of a symbolical part, where the storage scheme of the resulting matrix is created, and a numerical part, where the actual calculations are carried out (Pissanetzky 1984). The symbolical part is carried out only once at the beginning of the analysis, whereas the numerical part is evaluated repeatedly, but only for non-vanishing factors. For example, the numerical part of the product in Eq. (25) indeed only implies the operations described in Equations (26) and (27).

(2) Here, columns of $\hat{\varphi}_{B_\bullet}$ which are at the same time outputs of φ_A are removed.

(3) By this operation, rows of $\hat{\varphi}_{A_\bullet}$ which are at the same time outputs of φ_B should be removed. However, because no two mechanical components can contain identical output variables, $\hat{\varphi}_{A_\bullet}$ and I_{φ_B} do not share rows with nonzero elements and thus this operation is immaterial.

(4) At this place two sparse matrices $\hat{\varphi}_{B_\bullet} \circ \hat{\varphi}_{A_\bullet}$ and $\hat{\varphi}_{B_\bullet} \circ I_{\varphi_A}$ are added. However, the non-zero columns of $\hat{\varphi}_{B_\bullet} \circ \hat{\varphi}_{A_\bullet}$ must also be non-zero columns of $\hat{\varphi}_{A_\bullet}$, and, within these columns, the only non-zero elements are those corresponding to cases where an output (row) of φ_A matches an input (column) of φ_B. On the other hand, the only non-zero columns of $\hat{\varphi}_{B_\bullet} \circ I_{\varphi_A}$ are those which are also non-zero in $\hat{\varphi}_{B_\bullet}$, but do not correspond to the outputs of φ_A. Thus, there are no common non-zero elements between $\hat{\varphi}_{B_\bullet} \circ \hat{\varphi}_{A_\bullet}$ and $\hat{\varphi}_{B_\bullet} \circ I_{\varphi_A}$, so this term also does not give rise to "true" operations.

(5) Again, two sparse matrices $\hat{\varphi}_{B_\bullet} \circ (I_{\varphi_A} + \hat{\varphi}_{A_\bullet})$ and $\hat{\varphi}_{A_\bullet}$ are formally added. However, the outputs (rows with non-zero elements outside of the diagonal) of $\hat{\varphi}_{B_\bullet} \circ (I_{\varphi_A} + \hat{\varphi}_{A_\bullet})$ correspond to the outputs of $\hat{\varphi}_{B_\bullet}$, which can not be at the same time outputs of $\hat{\varphi}_{A_\bullet}$. Thus, $\hat{\varphi}_{B_\bullet} \circ (I_{\varphi_A} + \hat{\varphi}_{A_\bullet})$ and $\hat{\varphi}_{A_\bullet}$ do not share rows with nonvanishing elements, and this operation is also free of numerical effort.

Thus, only the "kernel" operation (1) involving the very few nonzero entries outside of the diagonals consumes computation time and it is possible to design automatic schemes which have the same efficiency as hand-coded methods. A further appealing property of this approach is that it fits neatly into the "client/server" model introduced above, and that it is not dependent on internal representation details. Indeed, all that is needed for its implementation is the support of two further functions "do_symbolics" and "do_numerics" in extension to the basic functions of kinetostatical transmission elements.

3.4. FORWARDS- AND BACKWARDS-PROPAGATION

The general scheme for the calculation of Jacobians still leaves some topics open concerning the eventual form of the operations. By studying all methodologies applied in robotics for the evaluation of Jacobians, it turns out that three basic features determine the type of a particular algorithm. These are (1) the reference point for the linear velocity, (2) the frame of decomposition for the vectors, and (3) the sequence in which multiplication is carried out. For the first two issues, typical selections are (a) the inertial frame $\mathcal{K}_0$, (b) the body-fixed frame $\mathcal{K}_i$ and (c) an intermediate frame $\mathcal{K}_\ell$ located between the inertial and end-effector frame. For the third choice mainly two alternatives are regarded: the "*Backwards*"-propagating multiplication scheme

$$\varphi_*^{\mathrm{B}} = \Big(\big((\varphi_{n*} \circ \varphi_{n-1*}) \circ \varphi_{n-2*}\big) \circ \cdots\Big) \circ \varphi_{1*} \tag{34}$$

and the "*Forwards*"-propagating scheme

$$\varphi_*^{\mathrm{F}} = \varphi_{n*} \circ \Big(\cdots \circ (\varphi_{3*} \circ (\varphi_{2*} \circ \varphi_{1*}))\Big) \ . \tag{35}$$

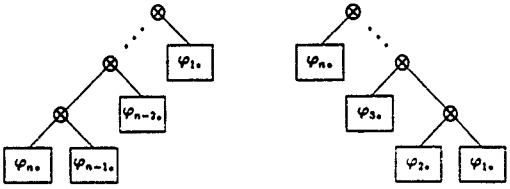
In the backwards-propating scheme, computations are basically performed from the tips of the branches of the system to the inertial frame. Only the spatial transformations between the tips and the intermediate frames ${}^k\mathrm{X}_i$ are evaluated here, and the local Jacobians are then transformed in one step to the frames at the tips. In contrast to this, in the forward-propagating schems also all spatial transformations between the inner frames, as well as the representations of the Jacobians in all subsequent frames are evaluated. This is shown in Table 1, where one may assume that $\mathcal{K}_k$ is an end-effector frame and $\mathcal{K}_1$ is a base frame. A good vehicle for the systematic treatment of the multiplication schemes is the

	${}^k\mathrm{X}_j$	${}^k\mathcal{J}_k^{(j)}$	Index sequence
φ_*^{B}	${}^k\mathrm{X}_{j+1}\ {}^{j+1}\mathrm{X}_j$	${}^k\mathrm{X}_{j+1}\ {}^{j+1}\mathcal{J}_{j+1}^{(j)}$	$j = k-2, \ldots, 1$
φ_*^{F}	${}^\kappa\mathrm{X}_{\kappa-1}\ {}^{\kappa-1}\mathrm{X}_j$	${}^\kappa\mathrm{X}_{\kappa-1}\ {}^{\kappa-1}\mathcal{J}_{\kappa-1}^{(j)}$	$j = 1, \ldots, k-2$ $\kappa = j+2, \ldots, k$

Table 1: Intermediate terms of the forwards- and backwards propagation schemes

concept of the *binary tree* (Knuth 1973), in which each node has either exactly two or no successors. Nodes without successors represent the "*leaves*" of the trees, while a regular

node represents an intermediate expression resulting from the product of the left and the right successor. In the setting of the evaluation of Jacobians, the leaves correspond to the operations of "globalizing" given local Jacobians, while the regular nodes represent the intermediate products in a sequence of transmission elements. In particular, the backwards- and forwards-propagating schemes have a very simple representation as a "left-biased" and a "right-biased" binary tree, respectively (Fig. 8).



a) backwards-propagating scheme b) forwards-propagating scheme

Figure 8: Binary-tree structures for basic multiplication schemes

4. Results

4.1. KINEMATICS

The discussed scheme for Jacobian generation has been applied to the kinematics of an unbranched chain of revolute and prismatic joints, as it is typical in robotics. Of the different possibilities of the combinations of reference point, decomposition frame and order of multiplication, five correspond to known approaches from the literature, whereas two are new and introduced here for the first time (Table 2). Another five combinations (the empty fields) are not further analyzed, as their practical benefit is not evident. This is because the reference frame $\mathcal{K}_i$ of Renaud 1981 lies approximately in the middle of the chain, from where a forwards-propagating composition is started towards the ends of the two resulting subchains, and a combination with representations in $\mathcal{K}_0$ or $\mathcal{K}_i$ is too complicated.

For the methods described in Table 2, a comprehensive operation count was performed in Krupp 1992. Here, only the results for a serial chain of n revolute joints — intermixed with corresponding rigid links — is summarized in Table 3. For the calculation, the well-known DENAVIT-HARTENBERG-parametrization for rigid-body motion was applied. The results are given both for the case that only the Jacobian of the end-effector (the frame at the end of the chain) or also of every frame at the base of each link is needed. The numbers in parenthesis correspond to a chain with $n = 6$. One can appreciate that, besides the method of Waldron 1981, every method for calculating all intermediate Jacobians involves $O(n^2)$ operations. Also, one can see that for the forwards-propagating methods, the evaluation of the end-effector Jacobian affords the same number of operations as calculating all intermediate Jacobians as well. Finally, note that Method I is most efficient when all

			Reference point for velocity		
			$\mathcal{K}_i$	$\mathcal{K}_0$	$\mathcal{K}_i$
Frame of decomposition	$\mathcal{K}_i$	F	Vukobratović/ Potkonjak 1979	Method I	
		B	Orin/ Schrader 1983	Method II	
	$\mathcal{K}_0$	F	Olson/ Ribble 1982	Waldron 1981	
	$\mathcal{K}_i$				Renaud 1981

F: fowards-propagation B: backwards-propagation

Table 2: Basic methods for Jacobian evaluation

intermediate Jacobians and a decomposition in the corresponding body-fixed frames is needed. Similarly, Method II is most efficient when only the Jacobian of the end-effector and a decomposition of vectors in this frame is needed. This shows that the new Methods I and II may be interesting for dynamics, because in this case the costly transformation of inertia properties to the inertial frame is avoided.

Method	Vukobratović/ Potkonjak 1979	Orin/ Schrader 1983	Olson/ Ribble 1982	Waldron 1981
only end-effector	$19n^2+24n+4$ (832)	$76n-34$ (422)	$76n-17$ (439)	$76n-126$ (330)
all frames	$19n^2+24n+4$ (832)	$53n^2+25n+26$ (1372)	$6n^2+72n-17$ (631)	$76n-126$ (330)

Method	Renaud 1981	Method I	Method II
only end-effector	$76n-193$ (263)	$13n^2+36n-9$ (675)	$76n-40$ (416)
all frames	not supported	$13n^2+36n-9$ (675)	$24n^2+55n-40$ (1154)

Table 3: Numerical effort of the basic Jacobian evaluation schemes

4.2. DYNAMICS

As an application of the described method to the problem of generation of the equations of motion, consider a branched system of links hinged to each other by revolute joints, where a "shaft" with two joints is connected to the inertial frame, at the end of which five further serial chains are attached, four of them having three, and one of them having four degrees of freedom. For this system, three basic methods for generation of the dynamical equations where applied: (a) the *composite rigid-body method* of Walker and Orin 1982, (b) the *articulated-body method* of Featherstone 1987, and (c) the Jacobian-based method described for example in Nikravesh 1988 Out of these, only the third variant shall be shortly described, as the other two are most well known in the dynamics and robotics community.

For each rigid body B_i, its inertia properties can be assumed to be given with respect to an arbitrary point P_i. Denoting by $\underline{t}_{P_i}$ the velocity twist of B_i at P_i and by $\underline{w}^*_{P_i}$ the resulting wrench of all applied forces and torques at B_i with respect to P_i, the principle of virtual velocities (Kane and Levinson 1985) is

$$\sum_{i=1}^{n_B} [(\Xi_{P_i} \dot{\underline{t}}_{P_i} + \underline{b}_{P_i}) - \underline{w}^*_{P_i}]^T \delta \underline{t}_{P_i} = 0 \ , \tag{36}$$

where n_B is the number of bodies, and Ξ_{P_i} and $\underline{b}_{P_i}$ are the "*spatial*" analogons of the tensor of inertia and the vector of coriolis and centrifugal forces, defined as

$$\Xi_{P_i} = \begin{bmatrix} \Theta_{P_i} & m_i \widetilde{{}_{P_i}r_{S_i}} \\ -m_i \widetilde{{}_{P_i}r_{S_i}} & m_i I_3 \end{bmatrix} \ , \ \underline{b}_{P_i} = \begin{bmatrix} \omega_i \times \Theta_{P_i} \omega_i \\ m_i \omega_i \times (\omega_i \times {}_{P_i}r_{S_i}) \end{bmatrix} \ , \tag{37}$$

and ${}_{P_i}r_{S_i}$ is the radius vector from P_i to the center of gravity S_i of B_i. By substituting

$$\underline{t}_{P_i} = J_{P_i} \underline{\dot{q}} \ , \ \delta \underline{t}_{P_i} = J_{P_i} \delta \underline{q} \ , \tag{38}$$

where $\underline{q} = [q_1, \ldots, q_f]^T$ represent the (independent) generalized coordinates, and taking into consideration the independency of virtual displacements $\delta \underline{q}$, the equations of motion in minimal coordinates

$$M(\underline{q}) \ddot{\underline{q}} + \underline{k}(\underline{q}, \dot{\underline{q}}) = \underline{Q}(\underline{q}, \dot{\underline{q}}) \ . \tag{39}$$

result, where

$$M(\underline{q}) = \sum_{i=1}^{n_B} J^T_{P_i} \Xi_{P_i} J_{P_i} \ , \ \underline{k}(\underline{q}, \dot{\underline{q}}) = \sum_{i=1}^{n_B} J^T_{P_i} (\Xi_{P_i} J_{P_i} \dot{\underline{q}} + \underline{b}_{S_i}) \ , \ \underline{Q}(\underline{q}, \dot{\underline{q}}) = \sum_{i=1}^{n_B} J^T_{P_i} \underline{w}^*_{P_i} \ . \tag{40}$$

With the notation

$$g = \text{diag} \left\{ \Xi_{P_1}, \ldots, \Xi_{P_{n_B}}, 0, \ldots, 0 \right\} \ , \tag{41}$$

$$\dot{\underline{b}} = [\Xi_{P_1} J_{P_1} \dot{\underline{q}} + \underline{b}_{S_1} - \underline{w}^*_{P_1}, \ldots, \Xi_{P_{n_B}} J_{P_{n_B}} \dot{\underline{q}} + \underline{b}_{S_{n_B}} - \underline{w}^*_{P_{n_B}}, 0, \ldots, 0]^T \ , \tag{42}$$

the complete set of equations can be written compactly as

$$\varphi_{n*}^T \, g \, \varphi_{n*} + \varphi_{n*}^T \, \dot{\underline{b}} = 0 \ . \tag{43}$$

The implementation of all three methods was easily possible based on our existing object-oriented programming environment for multibody systems named "MEHBILE".

Method	Time[1] [s]
composite rigid-body	0.58
articulated body	0.52
Jacobian-based	0.51

[1] On an Apollo DN 3500

Table 4: Computation time for one set of equations of motion

Table 4 shows the resulting computation time for generation and resolution (determination of generalized accelerations $\ddot{\underline{q}}$) of one complete set of equations of motions. In this table, not

all possible optimizations (special treatment of first and last bodies in the chain, elimination of operations involving zero coefficients contained in the local Jacobians) where carried out. However, all methods where treated similarly. Clearly, the Jacobian method is the most efficient one, even though the gain as compared to the other two methods is very small. However, for systems containing many closed multibody loops, the advantage of using the Jacobian-based method for generation of equations of motion will in general be higher.

5. Conclusions

The proposed method for modelling Jacobians in multibody systems establishes a unified and general formulation which is also well-suited for object-oriented programming techniques. The underlying differential-geometric interpretation gives a means of extending the concept of kinetostatic transmission to more general notions while at the same time hiding the implementation details from the user. This renders an "easy-to-grasp" intuitive interface, which is well-suited for rapid prototyping of mechanisms and complex multibody systems. The novel sparse matrix storage scheme introduces an alternative approach for storing the interconnection structure of multibody systems which is "nearer" to the formula level then other graph-theoretic methods. Finally, two new computation variants for the evaluation of Jacobians are presented, which consume less operations than those published previously in the case that all vector quantities are decomposed in the body-fixed frames. This makes the approach interesting for multibody dynamics, a topic which shall be discussed in more detail in the near future.

References

Andrews, G. C. and Kesavan, H. K. (1975). The vector network model: A new approach to vector dynamics. *Mechanism and Machine Theory*, 10:509–519.

Booch, G. (1991). *Object Oriented Design with Applications*. The Benjamin/Cummings Series in Ada and Software Engineering. Benjamin/Cummings Publishing Company, Redwood City, California.

Brandl, H., Johanni, R., and Otter, M. (1986). A very efficient algorithm for the simulation of robots and similar multibody systems without inversion of the mass matrix. In *IFAC/IFIP/IMACS Symposium on Robotics*, Wien.

Chace, M. and Smith, D. (1971). DAM—Digital computer program for the dynamic analysis of generalized mechanical systems. *SAE Paper No. 710244*.

Choquet-Bruhat, Y. and DeWitt-Morette, C. (1989). *Analysis, Manifolds and Physics. Part I: Basics*. North-Holland, Amsterdam, New York.

Featherstone, R. (1987). *Robot Dynamics Algorithms*. Kluwer Academic Publishers, Boston, Dordrecht, Lancaster.

Garcia de Jalón, J., Unda, J., Avello, A., and Jiménez, J. (1986). Dynamic analysis of three-dimensional mechanisms in "natural coordinates". ASME-Paper 86-Det-137.

Hiller, M. and Kecskeméthy, A. (1989). Equations of motion of complex multibody systems using kinematical differentials. *Transactions of the Canadian Society of Mechanical Engineers*, 13(4).113–121.

Hunt, K. H. (1990). *Kinematic Geometry of Mechanisms*. The Oxford Engineering Science Series. Clarendon Press, Oxford, paperback edition.

Kane, T. R. and Levinson, D. A. (1985). *Dynamics: Theory and Applications.* McGraw-Hill Series in Mechanical Engineering. McGraw-Hill Book Company, New York.

Kecskeméthy, A. (1993). *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen.* PhD thesis, Universität - GH - Duisburg.

Kecskeméthy, A. and Hiller, M. (1992). Automatic closed-form kinematics-solutions for recursive single-loop chains. In *Flexible Mechanisms, Dynamics, and Analysis, Proc. of the 22nd Biennal ASME-Mechanisms Conference, Scottsdale (USA)*, pages 387–393.

Kecskeméthy, A. and Hiller, M. (1993). Object-oriented approach for an effective formulation of multibody dynamics. In *Second U.S. National Congress on Computational Mechanics, August 16–18*, Washington D.C.

Knuth, D. E. (1973). *The Art of Computer Programming. Volume 1: Fundamental Algorithms.* Addison-Wesley Publishing Company, second edition.

Krupp, T. (1992). Objektorientierte Erstellung der Jacobimatrizen von Mehrkörpersystemen durch Verkettung dünnbesetzter Teilabbildungen. Master's thesis, Universität Duisburg, Fachgebiet Mechatronik.

Luh, J. and Zheng, Y. (1985). Computation of input generalized forces for robots with closed kinematics chain mechanisms. *IEEE Journal of Robotics and Automation*, RA-1(2):95–103.

Nikravesh, P. E. (1988). *Computer-Aided Analysis of Mechanical Systems.* Prentice-Hall.

Orlandea, N., Chace, M., and Calahan, D. (1979). A sparsity-oriented approach to the dynamic analysis and design of mechanical design of mechanical systems — Parts 1 & 2. *Transactions of the ASME, Journal of Engineering for Industry*, pages 773–784.

Pissanetsky, S. (1984). *Sparse Matrix Technology.* Academic Press.

Renaud, M. (1981). Geometric and kinematic models of a robot manipulator: Calculation of the Jacobian matrix and its inverse. In *Proceedings of the 11th International Symposium on Industrial Robots*, Tokyo.

Sheth, P. N. and Uicker Jr., J. J. (1972). IMP (Integrated Mechanisms Program), A computer-aided design analysis system for mechanisms and linkage. *Transactions of the ASME, Journal of Engineering for Industry*, pages 454–464.

von Westenholz, C. (1981). *Differentiel Forms in Mathematical Physics*, volume 3 of *Studies in Mathematics and its Applications.* North-Holland, Amsterdam, New York, Oxford, revised edition.

Waldron, K. (1981). *Geometrically based manipulator rate control algorithms.* PhD thesis, Ohio State University.

Walker, M. and Orin, D. (1982). Efficient dynamic computer simulation of robotic mechanisms. *Journal of Dynamical Systems, Measurement and Control*, 104:205–211.

Wirfs-Brock, R. and Wilkerson, B. (1989) Object-oriented design: A responsability-driven approach. In *OOPSLA '89 Proceedings*, pages 71–75.

Wittenburg, J. (1977). *Dynamics of Systems of Rigid Bodies*, volume 33 of *Leitfäden der angewandten Mathematik und Mechanik.* B.G. Teubner, Stuttgart.

Modeling of Multibody Systems with the Object-Oriented Modeling Language Dymola

Martin Otter
Institute for Robotics and System Dynamics, German Aerospace Research Establishment Oberpfaffenhofen (DLR), D-8031 Wessling, Germany, e-mail: df43@vm.op.dlr.de

Hilding Elmqvist
DynaSim AB, Research Park Ideon, 223 70 Lund, Sweden e-mail Elmqvist@Gemini.ldc.lu.se

François E. Cellier
Department of Electrical and Computer Engineering, The University of Arizona Tucson Arizona 85721 U.S.A., e-mail Cellier@ece.arizona.edu

ABSTRACT. The object-oriented modeling language Dymola allows the physical modeling of large interconnected systems based on model components from *different engineering domains*. It generates *symbolic* code for different target simulators. In this paper, a Dymola class library for the efficient generation of the equations of motion for multibody systems is presented. The library is based on a reformulated O(n) algorithm such that the algorithm is truly object-oriented. This feature can also be interpreted as a bond graph oriented modeling of multibody systems. Furthermore a new algorithm for a certain class of *variable structure* multibody systems is presented, which allows the generation of efficient symbolic code.

1 Introduction

Dymola [6, 8, 3, 4] is an object-oriented modeling language for modeling of large dynamical systems. Models are hierarchically decomposed into submodels which are connected in accordance with the *physical coupling* of the components. The features of Dymola allow the development of domain specific class-libraries for e.g. electronic circuits, control systems, hydraulic systems, thermodynamics systems, bond graphs and others, which can all be used in conjunction for generating a specific multi-domain application model. Dymola generates efficient *symbolic* code for several target simulators[1]. It can handle ordinary differential equation models as well as differential-algebraic equation (DAE) models. If a DAE is of higher index, certain parts of the equations are symbolically differentiated according to the algorithm of Pantelides [22].

[1]Presently, Dymola supports ACSL[18], DESIRE[13], SIMNON[7], SIMULINK[16] and plain Fortran in either Simnon- or DSblock-format[20].

In this paper it is shown how variable structure multibody systems can be modeled by Dymola. For this purpose a Dymola library based on the recursive O(n) algorithm of [1] is explained in detail. The algorithm had to be slightly modified in order to arrive at a "truly" object-oriented formulation as encouraged by Dymola. "Truly" object-oriented means, that physical objects of the multibody system are mapped to corresponding Dymola objects, which are connected in accordance with the physical coupling of the system (this is *no* block-diagram representation of input/output blocks). The class description of an object contains the equations that describe the object, and the *cut*-definition, i.e., a definition of the interface of that object to other objects. Due to this procedure, the equations of motion of a multibody system are determined by Dymola using only the connection structure of objects and the *local* information about objects. It turns out that this description form has close connections to the bond graph methodology [26, 3]. In fact, this object-oriented definition of a multibody system can be interpreted as a multibody bond graph[2].

Besides a reformulation of an O(n) algorithm, a new enhancement of this algorithm is presented that allows the efficient treatment of a certain class of variable structure systems, i.e., systems with varying degrees of freedom. In the last years, methods to handle multibody systems with variable structure have been considerably improved [14, 24, 11, 10]. Usually, such systems are modeled by *numerical* multibody programs. This is due to the fact that, if n independent switches are present to remove or add one degree of freedom, 2^n different configurations are possible. For example, if dry friction is present in the joints of a 6 degree-of-freedom (dof) robot, $2^6 = 64$ configurations are possible, since every joint can either be in sliding (= 1 dof) or in sticking (= 0 dof) mode. *Numerical* multibody programs are designed to handle a large class of multibody systems with the same program. If such a feature is present, it is not principally difficult to change the configuration (i.e., the multibody system) during integration. On the other hand, a *symbolic* multibody program generates code for a *specific* multibody system only. Therefore, 2^n different codes have in general to be generated, if there are 2^n possible configurations (= multibody systems). Even for a modestly sized multibody system, such as a 6 dof robot with dry friction, this would be quite impractical. It will be shown that a simple modification of the recursive O(n) algorithm circumvents this difficulty and will allow the *symbolic* generation of compact and efficient code for variable structure systems.

Dymola together with its multibody library is comparable to commercially available multibody programs both in terms of efficiency and ease-of-use. However as already noted, Dymola can easily model components from other engineering domains in conjunction with the multibody components by invoking them from other already available class libraries. For example, a sophisticated library for electronic components corresponding to the SPICE electronics program [19] (diodes, Zener diodes, tunnel diodes, BJTs, JFets, MosFets, and GaAs transistors) is being developed at the University of Arizona. In contrast to other multibody programs, Dymola supports *multidisciplinary* modeling within *one* environment — the multibody part being just one model component among other equally important engineering disciplines.

[2]Someone not familiar with the bond graph methodology can just skip related paragraphs in this paper

2 Detailed robot model

Dymola is introduced by means of a detailed dynamic model of the industrial robot Manutec r3 described in [9]. The structure of this "multidisciplinary" model is shown in Figure 1. The 6 degree-of-freedom robot consists of a system of rigid bodies connected by ideal revolute joints. Every joint is driven by a torque, produced by the electro-magnetic field of a current-controlled DC-motor and transformed by gear-boxes. The motors are controlled by decentralized cascade controllers. The block "rotor+gear" in Figure 1 contains the mechanical part of a motor and of its gear-box.

Dymola supports a *hierarchical decomposition* of models. The Dymola model can therefore be designed such that it directly reflects the model structure as shown by the following definition of the robot in the Dymola model language ("..." indicates similar items that have been omitted in order to shorten the text):

```
@R3robot.lib   {use library R3robot.lib}
model R3robot
    submodel (R3control) c1, c2, c3, c4, c5, c6
    submodel (R3motor)  m1 (R=110,R=400,C=1.E-6,...), ..., m6(...)
    submodel (R3gearA)  g1  (c=43,d=0.005,p=p1,...), g2(...), g3(...)
    submodel (R3gearB)  g4  (p=p4,...), g5(p=p5...), g6(p=p6,...)
    submodel (R3mbs)    robot  {multibody system}

    constant p1=-165, p2=210, p3=60, p4=-99, p5=79.2, p6=-99
    input q1d, q2d, q3d, q4d, q5d, q6d     {desired arm angles}
    input w1d, w2d, w3d, w4d, w5d, w6d  {desired arm angular velocities}

    connect c1 to m1 to g1 to robot.j1
        ...
    connect c6 to m6 to g6 to robot.j6

    {Controller input = gear-ratio * model-input}
    c1.qd  =  p1*q1d
    c2.qd  =  p2*q2d
        ...
    c6.wd  =  p6*w6d
end
```

According to the different block component types of Figure 1, model classes are defined in a library that is made available to the model (object) definition through the command *@R3robot.lib* (the @ operator tells Dymola to *include* the file *R3robot.lib*). New models (or objects) of these class definitions are instantiated by the command:

submodel *(class-name) object-name (parameters) ...*

For example the statement "submodel *(R3control) c1, c2, c3, c4, c5, c6*" instantiates six identical objects of the same class *R3control*. This is meaningful since, in the R3

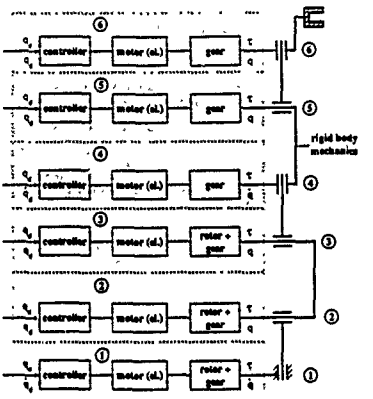


Figure 1: Model structure of Manutec r3 robot

robot, six identical cascade controllers are used. In contrast, the six motors of the robot have the same *structure* but different parameters. Therefore the class *R3motor* is used to describe the motor structure, whereas a specific motor is defined by supplying appropriate values for the motor parameters. Since the robot employs gear-boxes of two structurally different kinds, two different classes (*R3gearA*, *R3gearB*) are provided. Finally an object of the multibody description (i.e., the robot itself without the motors, gear-boxes and controllers) is instantiated from class *R3mbs*.

With the statement "constant *p1=−105,...*" the gear ratios p_i are defined as constants They are used in two different places (gear-box and input of controller). With the statements "input *q1d,...*" the input signals of the overall model are defined. These are the desired angles and angular velocities of the joints as input to the controllers.

All the objects are assembled by the connect statements. The meaning of a connect statement such as "connect *c1* to *m1*" is defined in the corresponding class description, as will be explained below. Note, that there is no signal *direction* associated with a connect statement. It is therefore *not* an input/output block-diagram description! Instead, the connect statement reflects the *physical* coupling of components. If an object has several different interfaces, also called cuts, the notation *object:cut* is used. For example, "connect *g1* to *robot:j1*" states that gear-box 1 is attached at cut *j1* (≈ joint 1) of object *robot*. At the end of the model description, the connections between the global input signals and the model are specified. For example, "*c1.qd = p1*q1d*" states that the input variable *qd* of controller *c1* is the desired angle *q1d* multiplied by the gear-ratio *p1*

To this point, the top most model components have been assembled. We shall now look at the component models themselves. Motors are described by objects of class *RSmotor*, which makes use of a basic Dymola library for electrical components such as resistors, capacitors, and operational amplifiers. Let us first have a look at one of the classes of this library in order to get familiar with some important concepts of Dymola. In the aforementioned library, a *capacitor* is defined as follows:

```
model class capacitor
    parameter C
    cut A (Va / i), B (Vb / -i)
    main cut cAB { A, B }
    main path pAB <A - B>
    local u

    Va - Vb = u
    C*der(u) = i
end
```

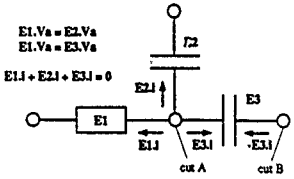


Figure 2: Dymola connection rules

A capacitor is an element with two cuts A and B, through which it can be connected to other elements. With each cut two variables are associated, the potential at the wire *(Va, Vb)* and the current flowing into the component through the wire *(i, −i)*. Dymola distinguishes between two types of variables: *across* and *through* variables. In a cut declaration, all variables to the left of the slash operator ("/") are defined to be *across* variables, whereas all variables to the right of the slash operator are defined to be *through* variables. Accordingly, the potentials *Va, Vb* are considered as *across* variables, whereas the current *i* is a *through* variable. The difference between these two variable types becomes apparent in a connection only. Assume for example, that three elements E1, E2, E3 with the above cut-definition are connected at one node as indicated in Figure 2. The Dymola built-in rules will generate the equations indicated in Figure 2, i.e., the *across* variables across a node are set equal, whereas the *through* variables through a node are summed up to zero. These are exactly the physical laws that apply when a capacitor is connected to other electrical elements. The statements "main cut ..." and "main path ..." in the capacitor definition state the default connections that are used if no cut-names are explicitly specified in the connect statement. For example, the statement "connect *C* at *(n1,n2)*" is equivalent to the longer specification "connect *C:A* at *n1*, *C:B* at *n2*.

The physical laws of the capacitor are programmed in the last two statements using the cut-variables and the der operator that characterizes a *differential*. An important feature of Dymola is that no *direction* is associated with variables. Based on the connection structure of the overall model and the problem description, Dymola will determine on its own for which variable each of the equations needs to be solved, and apply symbolic formula manipulation to transform the equations to the desired form, if necessary. Due to this feature, Dymola supports the use of true *equations* rather than simple assignment statements.

Let us now return to the robot example. Using the aforementioned library of basic electrical components, the motor shown in Figure 3 can be defined in the following way

```
model class R3motor
    submodel (resistor)   Rd1(R=200),Rd2(R=200),Rd3(R=100),Rd4(R=400)
    submodel (resistor)   Ri1(R=Ri1), Ri2(R=Ri2), Ri3(R=620), Rp(R=200), Ra(R=Ra)
    submodel (capacitor)  Ci(C=Ci)
    submodel (inductor)   La(L=La)
    submodel (emf)        emf(k=kmot)              {electro-motoric-force}
    submodel (hall)       hall(k=c)                {hall sensor}
    submodel (OpAmp)      diff, PI                 {operational amplifier}
    submodel (OpAmpP)     power                    {operational amplifier}
    submodel (ground)     g
    parameter Ra=400, La=1.E-3, kmot=3.45, Ri1=410, Ri2=500, Ci=1.E-7

    cut elec (Vd / id), mech (rq, rw), control (mech, elec)  {connection to controller}
    cut gear      (rq, rw / t)          {connection to gear}
    main cut c    (control, gear)       {default connection behaviour}
    main path p < control - gear >      {default connection behaviour}

    node    n0, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14
    connect Rd1 at (n2,n5),     Rd2 at (n1,n2),     Rd3 at (n3,n4),
            Rd4 at (n0,n3),     Ri1 at (n5,n6),     Ri2 at (n6,n7),
            Ri3 at (n0,n8),     Ci  at (n7,n9),     Rp  at (n0,n10),
            Ra  at (n11,n12),   La  at (n12,n13),   emf at (n13,n14),
            diff at (n4,n2,n5), PI  at (n8,n6,n9),  power at (n9,n10,n11),
            hall at (n14,n0,n3), gear at emf.mech,  elec  at n1, g at n0
end
```

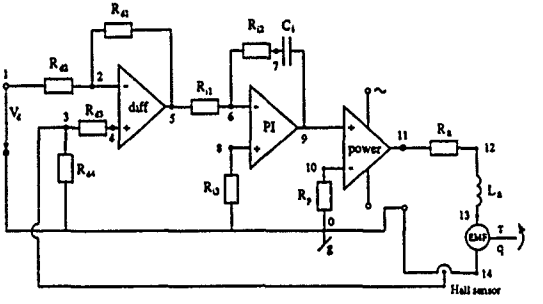


Figure 3: Model structure of Manutec r3 motors

The Dymola model reflects directly the underlying electrical circuit. Dymola supports a variety of different formats to define the connection of components. A statement of

the form "connect *Rd1* at *(n2,n5)*" means, that object *Rd1* is placed between the two nodes *n2*, *n5*. In the same way, the other classes *R3control*, *R3gearA* and *R3gearB* are defined, but they are omitted here due to space limitations. Class *R3mbs*, to describe the mechanical (multibody) part of the robot, is discussed after deriving the Dymola class library for multibody systems in the next two sections.

3 Basic classes of multibody library

The most difficult part in developing a Dymola library is the selection of basic objects as abstract mappings of their physical counterparts. Here, this selection is based on the object-oriented data model for multibody systems of [21]. The most fundamental object is a coordinate system, or *frame*, as shown in Figure 4 below. A frame i is always described with respect to a reference frame r ($= ref(i)$). If frame i is not moving with respect to

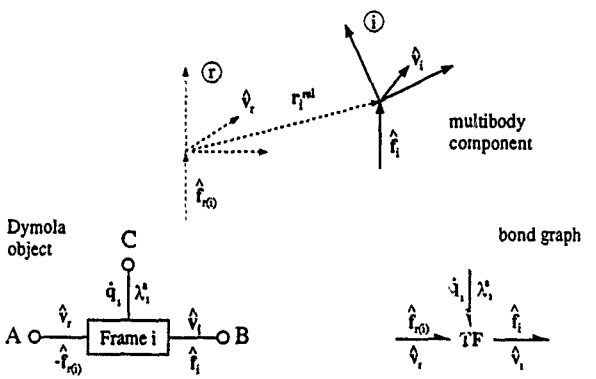


Figure 4: Fundamental Dymola class *Frame*

frame r, the two frames are material points located on the same rigid body. If there is a relative motion between the two frames, frame i describes an ideal joint, such as a revolute or spherical joint. The *inertial system* is the special frame *0*, where the movement of the frame is known (= is at rest) and is therefore the only frame, which has no reference frame. In the multibody library, class *Frame* describes the common properties of all different frame types. Specific frames, such as class *Revolute* describing a revolute joint, are derived by inheritance from superclass *Frame*.

The movement of frame i is uniquely described by the direction cosine matrix $\mathbf{T}_i$, which transforms tensors resolved in frame i into the inertial frame, and the absolute position vector $\mathbf{r}_i$, which is a vector pointing from the origin of the inertial frame to the origin of frame i. Equivalently, the movement of frame i on velocity level is described by the absolute angular velocity $\boldsymbol{\omega}_i$ of frame i and the absolute linear velocity $\mathbf{v}_i$ of the origin of this frame. If the movement of the reference frame r is known, the movement of frame i can be calculated, given the expressions for the relative motion.

At the origin of frame i, a force $\mathbf{f}_i$ and a torque $\boldsymbol{\tau}_i$ are acting. These two quantities result from cutting forces and torques as well as from internally applied forces and torques acting between frame i and frame r. For notational convenience, these two quantities are concatenated into a *generalized force vector* $\hat{\mathbf{f}}_i$, and equally are the angular and linear velocity of a frame concatenated into a *generalized velocity vector* $\hat{\mathbf{v}}_i$:

$$\hat{\mathbf{v}}_i = \begin{bmatrix} \boldsymbol{\omega}_i \\ \mathbf{v}_i \end{bmatrix} ; \qquad \hat{\mathbf{f}}_i = \begin{bmatrix} \boldsymbol{\tau}_i \\ \mathbf{f}_i \end{bmatrix}$$

With these preambles, Figure 4 can now be discussed in more detail. In the top part, frame i and its reference frame r are shown. Vector $\hat{\mathbf{v}}_i$ characterizes the movement of frame i, whereas vector $\hat{\mathbf{f}}_i$ characterizes the generalized force acting on frame i. Frame i can be viewed as a *transformer* element that transforms the "movement" $\hat{\mathbf{v}}_r$ of frame r to the "movement" $\hat{\mathbf{v}}_i$ of frame i and the force $\hat{\mathbf{f}}_i$ acting on frame i to force $\hat{\mathbf{f}}_{r(i)}$ acting with negative sign (due to the actio=reactio principle) on frame r using the usual transformation rules for velocities and forces.

The lower left part of Figure 4 depicts the corresponding Dymola object. This object has three *cuts* (A,B,C), which are used to connect the object to other objects. Cut C will be explained in more detail below. Every cut is a collection of several variables. The most important ones are shown in the figure. Note that there is no (input/output) direction associated with the variables of a cut. Dymola determines this information by itself, based on the connection structure and the problem description.

Two or more cuts can be connected at one node. As already noted, Dymola has two built-in rules for such a connection. *Across* variables are identical, whereas *through* variables sum up to zero. Due to the cut-principle in mechanics, all kinematic quantities of a cut (= position, velocity, acceleration) are treated as *across* variables, whereas the forces are treated as *through* variables. See for example Figure 6: Frame F_2 is the reference frame of frames F_3 and F_4. Cut B of frame F_2 and cut A of frames F_3, F_4 are therefore connected. The kinematic variables of these three cuts are identical since they describe the movement of the reference frame F_2. On the other hand, the forces sum up to zero, i.e., $\hat{\mathbf{f}}_2 - \hat{\mathbf{f}}_{2(3)} - \hat{\mathbf{f}}_{2(4)} = 0$.

Finally, the lower right part of Figure 4 shows the bond graph representation of a frame. Except for the (unusual) additional bond corresponding to cut C, a frame can be viewed as a *modulated transformer TF*, transporting the power $\hat{\mathbf{f}}_{r(i)}^T \hat{\mathbf{v}}_r$ from frame r to frame i without storing energy.

Until now, the description of class *Frame* was rather informal. To arrive at a rigorous specification by equations, some further definitions are necessary: Matrix $\mathbf{E}$ is the identity matrix. The coordinates of a vector $\vec{h}_i$, which is resolved in frame j, is given as a column vector ${}^j\mathbf{h}_i$. If $i = j$, index j can be removed, i.e., $\mathbf{h}_i$ is the vector $\vec{h}_i$ resolved in frame i. Additionally, operator *skew* and its inverse *vec* are defined as ($\mathbf{h}$ is a (3×1) column vector, $\mathbf{H}$ is a skew-symmetric (3×3) matrix):

$$\mathbf{H} = skew(\mathbf{h}) = skew\left(\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}\right) = \begin{bmatrix} 0 & -h_3 & h_2 \\ h_3 & 0 & -h_1 \\ -h_2 & h_1 & 0 \end{bmatrix} \; ; \quad \mathbf{h} = vec(\mathbf{H})$$

It is assumed that frame i has n_i degrees of freedom ($0 \leq n_i \leq 6$) with respect to frame r and that n_i generalized coordinates $\mathbf{q}_i(t)$ are used to uniquely describe the relative motion of frame i with respect to its reference frame r (e.g. for a frame describing a revolute joint, $\mathbf{q}_i$ is the rotation angle of the joint). Therefore, the relative direction cosine matrix $\mathbf{T}_i^{rel}(\mathbf{q}_i)$, describing the orientation of frame i with respect to frame r, and the relative position vector $\mathbf{r}_i^{rel}(\mathbf{q}_i)$, pointing from the origin of frame r to the origin of frame i, are functions of the generalized coordinates $\mathbf{q}_i$ only. This functional dependency is defined in the corresponding subclass of superclass *Frame*, e.g. in class *Revolute* to describe a revolute joint. Using these definitions, an object of class *Frame* is described by the following equations, utilizing the usual relations for relative kinematics, force transformations and d'Alembert principle (see, for example [25]), as well as a novel matrix $\mathbf{L}_i$ to define the variable structure of the system:

$$\begin{aligned}
\mathbf{T}_i &= \mathbf{T}_r \mathbf{T}_i^{rel^T} && \text{(1a)}\\
{}^0\mathbf{r}_i &= {}^0\mathbf{r}_r + \mathbf{T}_r\, {}^r\mathbf{r}_i^{rel} && \text{(1b)}\\
\hat{\mathbf{v}}_i &= \mathbf{C}_i \hat{\mathbf{v}}_r + \boldsymbol{\Phi}_i \dot{\mathbf{q}}_i && \text{(1c)}\\
\hat{\mathbf{a}}_i &= \mathbf{C}_i \hat{\mathbf{a}}_r + \boldsymbol{\Phi}_i \ddot{\mathbf{q}}_i + \boldsymbol{\xi}_i && \text{(1d)}\\
\hat{\mathbf{f}}_{r(i)} &= \mathbf{C}_i^T \hat{\mathbf{f}}_i && \text{(1e)}\\
\boldsymbol{\lambda}_i^a + \mathbf{L}_i \boldsymbol{\lambda}_i^l &= \boldsymbol{\Phi}_i^T \hat{\mathbf{f}}_i && \text{(1f)}\\
\mathbf{0} &= \mathbf{L}_i \ddot{\mathbf{q}}_i + (\mathbf{E} - \mathbf{L}_i)\boldsymbol{\lambda}^l && \text{(1g)}\\
P_i &= \hat{\mathbf{f}}_{r(i)}^T \hat{\mathbf{v}}_r + \boldsymbol{\lambda}_i^{a^T} \dot{\mathbf{q}}_i = \hat{\mathbf{f}}_i^T \hat{\mathbf{v}}_i && \text{(1h)}
\end{aligned}$$

where

$$\begin{aligned}
\boldsymbol{\Phi}_i(\mathbf{q}_i)\dot{\mathbf{q}}_i &= \hat{\mathbf{v}}_i^{rel} = \begin{bmatrix} \boldsymbol{\omega}_i^{rel} \\ \mathbf{v}_i^{rel} \end{bmatrix} = \begin{bmatrix} \mathbf{T}_i^{rel} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i^{rel} \end{bmatrix} \begin{bmatrix} vec\left(\mathbf{T}_i^{rel^T} \dot{\mathbf{T}}_i^{rel}\right) \\ {}^r\dot{\mathbf{r}}_i^{rel} \end{bmatrix} && \text{(2a)}\\
\mathbf{C}_i(\mathbf{q}_i) &= \begin{bmatrix} \mathbf{T}_i^{rel} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i^{rel} \end{bmatrix} \begin{bmatrix} \mathbf{E} & \mathbf{0} \\ -skew({}^r\mathbf{r}_i^{rel}) & \mathbf{E} \end{bmatrix} && \text{(2b)}\\
\boldsymbol{\xi}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i, \boldsymbol{\omega}_r) &= \begin{bmatrix} \mathbf{T}_i^{rel} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_i^{rel} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}_r \times {}^r\boldsymbol{\omega}_i^{rel} \\ \boldsymbol{\omega}_r \times (\boldsymbol{\omega}_r \times {}^r\mathbf{r}_i^{rel} + 2\,{}^r\mathbf{v}_i^{rel}) \end{bmatrix} + \frac{\partial \hat{\mathbf{v}}_i^{rel}}{\partial \mathbf{q}_i}\dot{\mathbf{q}}_i && \text{(2c)}\\
\mathbf{L}_i &= diag((L_{jj})_i), \quad (L_{jj})_i = \textit{if locked then } 1 \textit{ else } 0 && \text{(2d)}
\end{aligned}$$

Let us discuss these equations one after the other. Equation (1a) states, that the direction cosine matrix T_i of frame i is calculated using the direction cosine matrix of the reference frame r and the relative direction cosine matrix, which is only a function of q_i. In the same way, the absolute position vector ${}^0r_i$ of frame i, which is resolved in the inertial frame, is calculated using the absolute position vector of frame r and the relative position vector connecting the two frames.

The absolute position vector ${}^0r_i$ is resolved in the inertial frame, whereas all the other variables, e.g. velocities and forces, are resolved in the corresponding frames (i or r). The reason for this is that the position vector is e.g. needed in kinematic analysis problems or to determine animation data. In such problems, the vector has to be resolved in the inertial frame. On the other hand, velocities, accelerations and forces are utilized in dynamic analysis problems. For the most important of these applications, i.e., the direct kinetics problem, a detailed analysis of a certain class of multibody systems described in [12] unveils that the O(n) algorithm yields more efficient code when the calculation is performed in body-fixed coordinate systems rather than the inertial system. Furthermore, it is more natural to resolve forces in body-fixed frames since the directions of the forces can be associated with geometric properties of the corresponding component. Using the multibody library, it is of course easily feasible to resolve every vector in any desired coordinate system since the absolute direction cosine matrix of every frame is determined[3]

Equation (1c) states that the absolute velocity $\hat{v}_i$ of frame i, resolved in frame i is calculated using the absolute velocity of frame r and the relative velocity of frame i with respect to frame r. Equation (1d) states the same fact for the absolute acceleration $\hat{a}_i$ of frame i. These two equations have been derived by differentiating equations (1a,1b) twice, using the following definitions:

$$\hat{v}_i = \begin{bmatrix} vec\left(T_i^T \frac{d}{dt} T_i\right) \\ T_i^T \frac{d}{dt} {}^0r_i \end{bmatrix} \; ; \quad \hat{a}_i = \hat{T}_i^T \frac{d}{dt} {}^0\hat{v}_i \tag{3}$$

where

$$ {}^0\hat{v}_i = \hat{T}_i \hat{v}_i \; ; \quad \hat{T}_i = \begin{bmatrix} T_i & 0 \\ 0 & T_i \end{bmatrix}$$

Equation (1e) indicates how the force $\hat{f}_i$ acting on frame i is transformed to frame r. Note that, for this transformation, the same (but transposed) matrix C_i is used that is also utilized in the kinematic equations. Equation (1f) is an important equation relating forces and kinematics based on the d'Alembert principle. Neglecting the term $L_i\lambda_i^l$ for the moment, this equation states that the generalized force $\hat{f}_i$ acting on frame i and projected onto the free direction Φ_i of the frame is the generalized applied force λ_i^a acting between the two frames[4]. If no generalized applied force λ_i^a is present, $\Phi_i^T \hat{f}_i = 0$, i.e., $\hat{f}_i$ consists of $6 - n_i$ independent coordinates only — the constraint forces in the joint.

[3] The Dymola translator has an option, that *all* equations are removed which are not needed to compute the desired *output* variables. Therefore, equations (1a,1b) do not show up in the generated code if e.g. no animation data are needed and if no force elements are present.

[4] If the frame describes a revolute joint, for example, λ_i^a is the applied torque acting along the axis of rotation n_i, whereas $w_i = [n_i^T \; 0]^T$.

Equation (1g) is used to describe the variable structure of the frame in the following sense. It is assumed that every single degree-of-freedom (dof) of the frame can be locked, i.e., the corresponding generalized coordinates are fixed ($q_i = const$), whereas the first and second derivatives of these coordinates are set to zero. In such a situation, an additional constraint force λ_i^c has to be applied that acts in the same direction as the generalized applied force λ_i^a. The two possible states (free/locked) are defined by the diagonal matrix L_i. If the j-th diagonal element of L_i is one, the j-th dof is locked. If it is zero, the j-th dof applies. Therefore, equation (1g) states that $\dot{q}_i = 0$ if all degrees of freedom are locked, and that otherwise the constraint forces $\lambda_i^c = 0$. During integration, matrix L_i remains constant. The matrix may change its value only before the integration starts or immediately after an event has occurred.

Finally, equation (1h) states that the energy flowing into the frame[5] is the same as the energy flowing out of the frame since the frame does not store energy. This equation can easily be derived from equations (1c,1e,1f).

In Figure 4, the bond graph of a frame has an unusual bond, corresponding to cut C. The reason is that variables $\hat{f}_{r(i)}, \hat{v}_r, \hat{f}_i, \hat{v}_i$ and $\lambda_i^a, \dot{q}_i$ are related through equations (1). Since these relations hold for every single frame object, a bond graph would become very complex if this structure were always shown explicitly using basic bond graph elements. For simplicity, it was therefore decided to generalize the *transformer* element instead.

Usually, bonds transmit only effort and flow variables. This is not the case in the multibody bond graph. Here, the position (= integral of the flow variable) and the acceleration (= derivative of the flow variable) are also transmitted, due to equations (1a,1b,1d). There are two reasons for this decision: First of all, the integral of the flow variable does not always exist, since in a general three-dimensional movement, the angular velocity is not integrable to a position coordinate[6]. Secondly, nearly all multibody systems represent higher-index differential algebraic equations (DAE) if the connection of bond graph elements is done on velocity level only. The explicit usage of the additional position and acceleration variables can be viewed as an application of the general "dummy derivative" technique to reduce the index of a DAE [4, 17]. In this case, the equations (1a,1b) were manually differentiated and included as (1c,1d). Alternatively, the Pantelides algorithm [22] could have been used to determine what equations to differentiate in order to reduce the index.

It is straightforward to build a Dymola class description based on equations (1). For this, equations (1) are split into two parts. All terms depending on relative quantities (e.g r_i^{rel} or Φ_i) are defined in an appropriate subclass (e.g. class *Revolute* for revolute joints), whereas all other equations are defined in the common superclass *Frame*.

Class *Frame* allows a rather general description of joints. Also, *rheonomic* joints are included in the description although $T_i^{rel}(q_i)$ and $r_i^{rel}(q_i)$ do not explicitly depend on time. This is due to the fact that, in the above equations, it is *not* defined what is known

[5] Power P = energy flow

[6] Instead of the integral of the "critical" flow variable (= angular velocity), the closely related direction cosine matrix T_i is used, see equation (3)

and what is unknown. For example, an object of class *Revolute* is used as revolute joint if q_i is *unknown* whereas $q_i, \dot{q}_i, \lambda_i^a$ are known. On the other hand, an object of the same class is used as pure rheonomic joint if $q_i, \dot{q}_i, \ddot{q}_i$ are known functions of time whereas λ_i^a is unknown. The possibility to use the same class description for different problems is one of the most important strengths of the Dymola modeling language.

In order to be able to describe general multibody systems, a few additional simple classes are needed as shown in Figure 5. Class *Body* defines the mass properties of a rigid body. An object of this class has only one *cut*, which can be connected at either cut A or cut B of an object of class *Frame*. The frame at which an object of class *Body* is attached

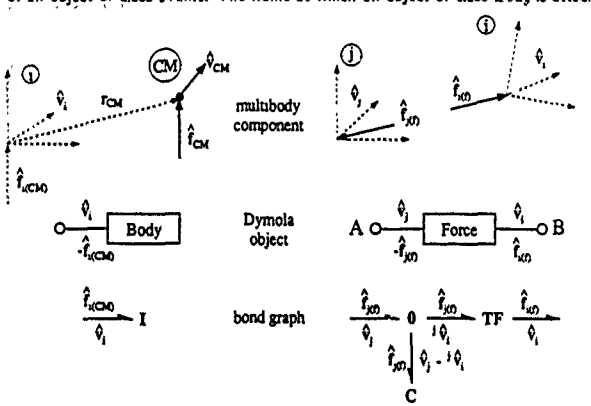


Figure 5: Dymola classes *Body* and *Force*

is usually located at the center of mass of the corresponding rigid body, but it can also be attached to any other point on the body. An object of this class is described by the following equations (see, for example [25]):

$$\hat{f}_{i(CM)} = \mathbf{I}_{i(CM)}\hat{a}_i + b_{i(CM)} \tag{4}$$

where

$$\mathbf{I}_{i(CM)} = \begin{bmatrix} \mathbf{I}_i' & m_i skew({}^i r_{CM}) \\ -m_i skew({}^i r_{CM}) & m_i \mathbf{E} \end{bmatrix} \tag{5a}$$

$$b_{i(CM)} = \begin{bmatrix} \omega_i \times \mathbf{I}_i' \omega_i \\ m_i \omega_i \times (\omega_i \times {}^i r_{CM}) \end{bmatrix} \tag{5b}$$

$$\mathbf{I}_i' = {}^i\mathbf{I}_{CM} - m_i skew({}^i r_{CM}) skew({}^i r_{CM}) \tag{5c}$$

Equation 4 incorporates the Newton-Euler equations, i.e., it states that the derivatives of the generalized momenta of a body are equal to the resultant generalized forces $\hat{f}_{(CM)}$ acting at the reference point used for the momentum balance. In Figure 5, the corresponding bond graph, i.e., the bond graph inertia element I, is shown.

Finally, class *Force* (cf. Figure 5) describes a force element. Similarly to class *Frame*, class *Force* is the superclass of all force elements containing their common properties. A specific force element, such as a spring, is a subclass of class *Force*. An object of this class always acts between two different frames and has therefore two cuts with which the object is connected to the two frames. The absolute position and the absolute velocity variables of these two frames are usually utilized to compute the relative position and the relative velocity variables, which in turn are used to formulate the force law, i.e., to calculate the applied force and torque acting on the two frames.

In the multibody library of Dymola, some additional classes are defined, e.g. class *Sensor* to measure kinematic quantities between two frames. Since these classes are not important in the context of this paper, they are not described here. An example of a multibody system is given in Figure 6. In the top part, an abstract picture of the multibody system is presented. The center part shows the corresponding Dymola objects together with their connection structure. The bottom part displays the bond graph of the multibody system.

The classes introduced so far are sufficient to describe rigid multibody systems with open and closed kinematic loops as well as arbitrary force elements. Depending on what are the known and unknown variables, Dymola can generate different types of simulation models from a general topological description of the multibody system. For example, if the movement of a multibody system is given and the driving forces λ_i^a in the joints are required, Dymola generates the *inverse model*. On the other hand, if the generalized accelerations $\ddot{q}_i$ have to be calculated, Dymola generates the *equations of motion* or *direct model*. For such a task, Dymola instantiates the equations for every involved object from the class library, adds the equations for the object connections, and *sorts* all equations in such a way that all unknown variables are calculated from the known variables. For the *inverse model*, it turns out that this strategy will generate equations to calculate the unknown joint forces in exactly the same way as the O(n) algorithm[7] of Luh/Walker/Paul [15]. The generated code is therefore very efficient.

In the case of the *equations of motion*, several practical problems appear when calculating the generalized accelerations of the system. First of all, the user has to select minimal coordinates, which may not be a trivial task if the system contains closed kinematic loops. Secondly, Dymola will detect a very large (sparse) system of equations that could be solved as a general system of index 1 differential-algebraic equations (DAE). However, since this system of equations is huge even for modestly-sized multibody systems, standard general-purpose DAE solvers such as DASSL [23] are not applicable. If only joint coordinates q_i are used as minimal coordinates and if the multibody system has no closed kinematic loops, the large system of equations is linear. Dymola can solve linear equations symbolically in

[7] n: number of degrees of freedom

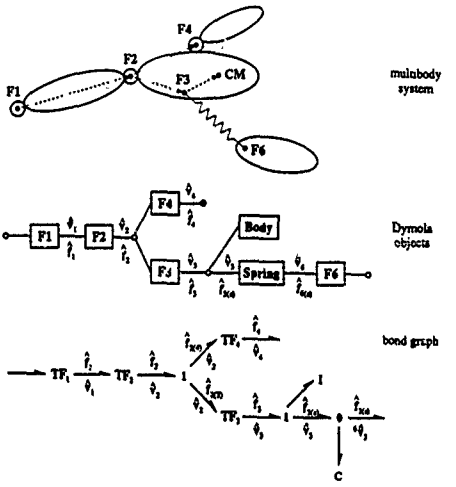


Figure 6: Example of multibody system modeling using Dymola

order to reduce the DAE to state space form. Several examples have indicated that the standard built-in method of Dymola to solve this matrix equation will usually produce very inefficient code. With special *tearing* mechanisms, it is possible to make the solution of the matrix equation more efficient. While Dymola supports tearing, an application of this technique requires some user interaction

In the next section, an alternative approach is explained. For the *direct problem*, a second class library called *mbsdir.lib* is provided, which has essentially the same interfaces as the class library *mbsinv.lib* explained above. A multibody system is defined in *exactly* the same way as before. However, the equations of the class-library *mbsinv.lib* have been transformed in such a way that Dymola will produce very efficient code for tree-structured multibody systems with variable structure.

4 Adaptation of class library to direct problem

Equations (1) form the starting point to reformulate the class library. The reformulation is based on the assumption that it is always possible to express the force and torque $\hat{f}_i$,

which is transferred between the cuts of multibody objects, as a linear function of the absolute linear and angular acceleration $\hat{\mathbf{a}}_r$ transferred via the same cut, i.e.,

$$\hat{\mathbf{f}}_i = \mathbf{I}_i \hat{\mathbf{a}}_r + \mathbf{b}_i \; ; \qquad \mathbf{I}_i = \mathbf{I}_i^T \tag{6}$$

This assumption holds true for classes *Body* and *Force* since the corresponding equation (4) of a body object is already in this form, and since the equation for a force object is a special case with $\mathbf{I}_i = 0$ because a force law does not depend on accelerations. What remains to be shown is that this assumption also holds for class *Frame*. For the proof of this statement, assume first that equation (6) holds true for cut B of an object of class *Frame*. According to Wehage [27], equations (1d,1f,1g) together with equation (6) can be combined to form the following linear, symmetric matrix equation:

$$\begin{bmatrix} \mathbf{I}_i & -\mathbf{E} & \mathbf{0} & \mathbf{0} \\ -\mathbf{E} & \mathbf{0} & \boldsymbol{\Phi}_i & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_i^T & \mathbf{0} & -\mathbf{L}_i \\ \mathbf{0} & \mathbf{0} & -\mathbf{L}_i & \mathbf{L}_i - \mathbf{E} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{a}}_i \\ \hat{\mathbf{f}}_i \\ \ddot{\mathbf{q}}_i \\ \boldsymbol{\lambda}_i^I \end{bmatrix} = \begin{bmatrix} -\mathbf{b}_i \\ -\mathbf{C}_i \hat{\mathbf{a}}_r - \boldsymbol{\xi}_i \\ \boldsymbol{\lambda}_i^a \\ 0 \end{bmatrix} \tag{7}$$

Equation (7) states that the unknown variables of frame i can be expressed as linear functions of the unknown acceleration $\hat{\mathbf{a}}_r$ of frame r. Solving (7) explicitly for the unknown quantities of frame i results in:

$$\boldsymbol{\lambda}_i^I = -\mathbf{L}_i^* \mathbf{h}, \tag{8a}$$

$$\ddot{\mathbf{q}}_i = \mathbf{M}_i^{-1}(\mathbf{h} + \boldsymbol{\lambda}_i^I) \tag{8b}$$

$$\hat{\mathbf{a}}_i = \mathbf{C}_i \hat{\mathbf{a}}_r + \boldsymbol{\Phi}_i \ddot{\mathbf{q}}_i + \boldsymbol{\xi}_i \tag{8c}$$

$$\hat{\mathbf{f}}_i = \mathbf{I}_i \hat{\mathbf{a}}_i + \mathbf{b}_i \tag{8d}$$

where

$$\mathbf{h} = \boldsymbol{\lambda}_i^a - \boldsymbol{\Phi}_i^T (\mathbf{I}_i (\mathbf{C}_i \hat{\mathbf{a}}_r + \boldsymbol{\xi}_i) + \mathbf{b}_i) \tag{9a}$$

$$\mathbf{M}_i = \boldsymbol{\Phi}_i^T \mathbf{I}_i \boldsymbol{\Phi}_i \tag{9b}$$

$$\mathbf{L}_i^* = \left(\mathbf{L}_i \mathbf{M}_i^{-1} + \mathbf{E} - \mathbf{L}_i\right)^{-1} \mathbf{L}_i \mathbf{M}_i^{-1} = \mathbf{L}_i \; \textit{if} \; dim(\mathbf{L}_i) = 1 \times 1 \tag{9c}$$

Inserting these equations into (1e) yields the transformed force $\hat{\mathbf{f}}_{r(i)}$ as a function of the acceleration of frame r:

$$\hat{\mathbf{f}}_{r(i)} = \mathbf{I}_{r(i)} \hat{\mathbf{a}}_r + \mathbf{b}_{r(i)} \; ; \qquad \mathbf{I}_{r(i)} = \mathbf{I}_{r(i)}^T \tag{10}$$

where

$$\mathbf{I}_{r(i)} = \mathbf{C}_i^T \mathbf{N}_i \mathbf{C}_i \tag{11a}$$

$$\mathbf{b}_{r(i)} = \mathbf{C}_i^T \left(\mathbf{b}_i + \mathbf{N}_i \boldsymbol{\xi}_i + \mathbf{I}_i \boldsymbol{\Phi}_i \mathbf{M}_i^{-1} (\mathbf{E} - \mathbf{L}_i^*)(\boldsymbol{\lambda}_i^a - \boldsymbol{\Phi}_i^T \mathbf{b}_i)\right) \tag{11b}$$

$$\mathbf{N}_i = \mathbf{I}_i - \mathbf{I}_i \boldsymbol{\Phi}_i \mathbf{M}_i^{-1} (\mathbf{E} - \mathbf{L}_i^*) \boldsymbol{\Phi}_i^T \mathbf{I}_i \tag{11c}$$

$$\mathbf{M}_i = \boldsymbol{\Phi}_i^T \mathbf{I}_i \boldsymbol{\Phi}_i \tag{11d}$$

$$\mathbf{L}_i^* = \left(\mathbf{L}_i \mathbf{M}_i^{-1} + \mathbf{E} - \mathbf{L}_i\right)^{-1} \mathbf{L}_i \mathbf{M}_i^{-1} = \mathbf{L}_i \; \textit{if} \; dim(\mathbf{L}_i) = 1 \times 1 \tag{11e}$$

It has thus been proven that (6) implies (10). The fact that forces at a connection node sum to zero, that accelerations are identical and that I_i and b_i are declared as through variables implies that (6) holds. Q.E.D.

The class library *mbsdir.lib* is built by using the already explained class library *mbsinv.lib* and by replacing equations (1d–1g) by equations (8–11). Furthermore, not the torque and force f_i is propagated through cuts, but the linear factors I_i, b_i of the torque and force in accordance with (6,10).

The main advantage of this reformulation lies in the fact that Dymola can sort the equations of a tree structured multibody system explicitly for the generalized accelerations $\ddot{q}_i$ without encountering algebraic loops (with the exception of small linear systems of equations within frame objects, which are solved by inverting matrix M_i of (11d)). This property can be explained as follows: The linear factors I_i, b_i are known at *Body*- and *Force*-objects. Using equations (11a,11b), these factors are propagated through all *Frame*-objects finally arriving at the inertial system. Since the acceleration of the inertial system is known, equations (8) can be used to calculate all unknown variables of the objects that are directly attached to the inertial frame. Afterwards the accelerations of these objects are known, and therefore the unknown variables of all objects attached to them can be calculated, and so on. Of course, this feature is only valid for tree-structured multibody systems.

The number of operations in the generated code is proportional to the number of degrees of freedom (= O(n)) of the multibody system and is therefore very efficient, especially if n is high. In the same sense as the inverse problem solution with library *mbsinv.lib* is equivalent to the Luh/Walker/Paul algorithm [15], the direct problem solution with library *mbsdir.lib* is equivalent to the recursive O(n) algorithm of Brandl/Johanni/Otter [1]. Note that these two algorithms are *not* explicitly programmed. Instead, only local properties of objects are stored in the corresponding class libraries. Due to the built-in connection rules of Dymola together with its sorting algorithm, Dymola "reinvents" these algorithms on its own.

Variable structure systems

The above class library extends the recursive algorithms such as [1] in one important aspect. it allows the handling of variable structure systems. As already noted, matrix L_i signals whether a degree of freedom of a joint is locked or not. As can be seen, L_i appears in some places in the recursive relations (11). The occurrences of matrix L_i can easily be interpreted: Assume that all degrees-of-freedom of a joint are free, i.e., $L_i = 0$, and therefore $L_i^* = 0$. In this case, (11) are the usual recursive relations On the other hand, if all degrees-of-freedom are locked, $L_i = E$ and therefore $L_i^* = E$, and equations (11) reduce to the recursive relations of a fixed joint with zero degrees of freedom The generated equations are nearly as efficient as if matrix L_i were not present. This can easily be seen for a joint with one degree of freedom. In this case, matrices M_i and L_i

are scalars, and only two multiplications, one subtraction and one addition are added in order to calculate $M_i^{-1}(E - L_i^*)$, $-L_i^* h_i, h_i + \lambda_i^l$, i.e., the efficiency reduction due to the variable structure is negligible.

The variable structure of a multibody system may be caused by a brake, by a stop, or by sticking friction. In the latter case, an additional problem appears since the sliding friction force is usually a function of the constraint forces in the joint. If this function is linear in the constraint forces, it is possible to generalize the above equations without destroying the recursive O(n) nature.

The force and torque $\hat{f}_i$ acting on frame i can explicitly be expressed as a function of the known applied forces λ_i^a and the unknown constraint forces λ_i^c in a form proposed by Roberson and Schwertassek [25]:

$$\hat{f}_i = \Psi_i^a(\lambda_i^a + L_i \lambda_i^l) + \Psi_i^c \lambda_i^c \tag{12}$$

Matrices $\Psi_i^a = \Psi_i^a(q_i), \Psi_i^c = \Psi_i^c(q_i)$ are defined such that the following relations hold

$$\Phi_i^T \Psi_i^a = E; \quad \Phi_i^T \Psi_i^c = 0 \tag{13}$$

If the joint is in sliding mode, the friction forces λ_i^a are linear functions of the constraint forces λ_i^c, i.e.,

$$\lambda_i^a = (E - L_i)\mu_i \lambda_i^c \tag{14}$$

Inserting equation (14) into equation (12) yields:

$$\hat{f}_i = \Psi_i^a(L_i \lambda_i^l) + \bar{\Psi}_i^c \lambda_i^c \tag{15}$$

where

$$\bar{\Psi}_i^c = \Psi_i^c + \Psi_i^a(E - L_i)\mu_i$$

In a second step, a matrix $\bar{\Phi}_i$ has to be determined, which lies in the null-space of $\bar{\Psi}_i^c$, i.e.,

$$\bar{\Phi}_i^T \bar{\Psi}_i^c = 0 \tag{16}$$

Finally, equation (1f) has to be replaced by (15,16). As before, the recursive relations are derived from a linear system of equations, which is equivalent to (7):

$$\begin{bmatrix} I_i & -E & 0 & 0 \\ -E & 0 & \Phi_i & 0 \\ 0 & \bar{\Phi}_i^T & 0 & -\bar{\Phi}_i^T \Psi_i^a L_i \\ 0 & 0 & -L_i & L_i - E \end{bmatrix} \begin{bmatrix} \hat{a}_i \\ \hat{f}_i \\ \ddot{q}_i \\ \lambda_i^l \end{bmatrix} = \begin{bmatrix} -b_i \\ -C_i \hat{a}_r - \xi_i \\ 0 \\ 0 \end{bmatrix} \tag{17}$$

Equation (17) has nearly the same structure as (7) with the major difference that Φ_i^T is replaced by matrix $\bar{\Phi}_i^T$. As a consequence, the system matrix of (17) is no longer symmetric. Hence, the linear factors I_i of the forces $\hat{f}_i$ are no longer symmetric either. Except for this property, the derivation proceeds as before, and the resulting equations are identical with the previous equations with the deviation that Φ_i^T is replaced by $\bar{\Phi}_i^T$ everywhere.

5 Dymola model of mechanical part of robot

The Dymola library for multibody systems as explained in the last two sections is not provided here due to space limitations. However, to demonstrate its usage, the multibody model of the 6 dof robot of section 2 is presented:

```
model class R3mbs
    submodel (Inertial)  i(ng3=-1)
    submodel (Revolute)  r1(n3=1), r2(n1=i), r3(n1=1,r3=0.5), ...
    submodel (Body)      b1(I33=1.16)
    submodel (Body)      b2(m=56.5, r1=0.172, r3=0.205,
                            I11=2.58, I22=2.73, I33=0.64, I31=-0.46)
                ...
    submodel (Body)      load(m=0, r1=0, r3=0, I11=0, I22=0, I33=0, I31=0)

    cut j1 (q1, qd1, qdd1 / t1)   {angle, angular velocity,
                                   angular acceleration / torque in joint}
    cut j2 (q2, qd2, qdd2 / t2)
                ...
    cut j6 (q6, qd6, qdd6 / t6)

    connect i to r1 to r2 to r3 to r4 to r5 to r6,
            b1 at r1, b2 at r2, b3 at r3, b4 at r4, b5 at r5, load at r6,
            j1 at r1:Axis, j2 at r2:Axis, j3 at r3:Axis,
            j4 at r4:Axis, j5 at r5:Axis, j6 at r6:Axis

    {body 6 without load is neglected with respect to rotor of body 6}
end
```

To define the robot, it is positioned in a special configuration (here: "vertical position" of all bodies). All vectors and tensors are given in this configuration, resolved in the inertial frame. A revolute joint r_i is for example defined by the position vector r_i from the reference frame r to frame i of the joint and the axis of rotation n. All coordinates of these vectors are zero by default. Therefore the declaration "submodel *(Revolute) r3(n1=1,r3=0.5)*" defines a revolute joint, which is located 0.5m in the direction of the 3-axis of frame r2 (due to "connect *r2* to *r3*") and which rotates about axis 1 of the new frame r3. When the axes of frames r2 and r3 coincide, the rotation angle is zero. The other multibody objects are defined in a similiar fashion. The interface of the robot to the outside world is given by cuts *j1*, ..., *j6*, which transfer the angle, the angular velocity, the angular acceleration and the torque in the revolute joints.

Generating the equations of motion and the inverse model for an object of the above class (i.e., the multibody part of the robot only), Dymola produces code for each of these problems within about 10 seconds[8] containing the following operational counts.

[8] on an Apollo workstation with a Motorola 68040 processor

	direct problem	inverse problem
*, / operations	727	269
+, - operations	493	190

The numbers of operations are approximately the same as for other symbolic multibody programs.

6 Conclusions

In this paper, a new library for the object-oriented modeling language Dymola was presented that supports the modeling of general multibody systems consisting of a connection of rigid bodies, ideal joints, and force elements. The presented library generates efficient code for tree-structured multibody systems. Multibody systems with closed kinematic loops, such as a five-point wheel suspension system with four closely coupled loops and one dof, have also been studied already, but the treatment of such systems is not yet sufficiently comfortable due to required user interaction. This situation will improve soon.

Dymola must not be viewed as just "yet another" multibody program. The unique feature of Dymola is its support for modeling components from several different engineering disciplines within *one* environment. For example, Dymola has been successfully applied to the modeling of the thermodynamic behavior of a house using bond graphs [28], chemical reaction systems [2], and electronic circuits. Presently, Dymola is enhanced by sophisticated new language elements for the definition and handling of discrete-event systems based on time and state events [5]. This will also add new functionality to the multibody library, since sampled data systems, interconnected friction elements and other discontinuous elements can be handled easily by use of these language extensions.

References

[1] Brandl H., Johanni R., and Otter M. A very efficient algorithm for the simulation of robots and similar multibody-systems without inversion of the mass-matrix. In Proc IFAC/IFIP International Symposium on Theory of Robots, Vienna/Austria, December 1986.

[2] Brooks B A., and Cellier F E.. Modeling of a Distillation Column Using Bond Graphs Proceedings SCS Int. Conf on Bond Graph Modeling San Diego California, pp 315-320. January 17-20, 1993

[3] Cellier F.E., Continuous System Modeling. Springer-Verlag, New York, 1991.

[4] Cellier F E., and Elmqvist H. Automated Formula Manipulation Supports Object-Oriented Continuous-System Modeling IEEE Control System Magazine 13(2), pp 28-38. 1993

[5] Cellier F E., Elmqvist H., Otter M., and Taylor J H. Guidelines for Modeling and Simulation of Hybrid Systems IFAC World Congress Sydney, Australia, July 1993

[6] Elmqvist H. A Structured Model Language for Large Continuous Systems Ph D dissertation Report CODEN LUTFD2/(TFRT-1015). Department of Automatic Control Lund Institute of Technology, Lund Sweden, 1978

[7] Elmqvist H., Åström K J., Schönthal T., and Wittenmark B.. *Simnon — User's Guide for MS-DOS Computers*, SSPA Systems, Gothenburg, Sweden, 1990

[8] Elmqvist H.: Dymola — User's Manual. DynaSim AB, Research Park Ideon, Lund, Sweden, 1993.

[9] Franke J., and Otter M.: The Manutec r3 Benchmark Models for the Dynamic Simulation of Robots "The International Handbook on Robot Simulation Systems", edited by D. Wloka, Volume I, to appear.

[10] Glocker C., and Pfeiffer F.: Complementary problems in multibody systems with planar friction. Accepted for publication in "Applied Mechanics", to appear.

[11] Haug E.J., Wu S.C., and Yang S.M.: Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact and Constraint Addition-Deletion – Part I, Theory Mechanism and Machine Theory, Vol. 21, pp. 401–506, 1986.

[12] Hiller M., Kecskeméthy A., and Stelzle W.: Ein Vergleich rekursiver Verfahren. Internal report, Universität Duisburg, Fachgebiet Mechatronik & Mechanik. A short version of this paper appeared in Stelzle W.: Ein Vergleich rekursiver Verfahren zur Untersuchung der Dynamik baumstrukturierter Mehrkörpersysteme ZAMM 73, pp. 107 – 109, 1993.

[13] Korn G.A.: Interactive Dynamic-System Simulation. McGraw-Hill, 1989.

[14] Lötstedt P.: Coulomb friction in two-dimensional rigid body systems. ZAMM 61, pp. 605–515 1981

[15] Luh J.Y.S., Walker M.W., and Paul R.P.C.: On-line computational scheme for mechanical manipulators. Trans. ASME Journal of Dynamic Systems Meas. and Control, Vol. 102, pp. 69-76, 1980

[16] Mathworks Inc.: SIMULINK – User's Manual. South Natick, Mass., 1992.

[17] Mattsson S.E., and Söderlind G.: A New Technique for Solving High-Index Differential-Algebraic Equations Using Dummy Derivatives. IEEE Symposium on Computer-Aided Control System Design, CACSD'92, Napa, California, March 1992.

[18] Mitchell E.E.L., and Gauthier J.S.: ACSL: Advanced Continuous Simulation Language – Reference Manual. Edition 10.0, MGS, Concord., Mass., 1991.

[19] Nagel L.W.: SPICE2: A computer program to simulate semiconductor circuits. Berkeley, University of California, Electronic Research Laboratory, ERL – M 520, 1975.

[20] Otter M.: DSblock: A neutral description of dynamic systems. Version 3.2. Technical Report TR R81-92, Institute for Robotics and Systemdynamics, DLR Oberpfaffenhofen, May 1992.

[21] Otter M., Hocke M., Daberkow A., and Leister G.: An Object-Oriented Data Model for Multibody Systems "Advanced Multibody System Dynamics", edited by W. Schiehlen, pp 19–58, Kluwer Academic Publishers, 1993.

[22] Pantelides C.C. The consistent initialization of differential-algebraic systems. SIAM Journal of Scientific and Statistical Computing, No. 9, pp. 213-231, 1988.

[23] Petzold L.R.: A description of DASSL: A differential/algebraic system solver Proc. 10th IMACS World Congress, Montreal, August 8-13, 1982.

[24] Pfeiffer F.: Über unstetige, insbesondere stoßerregte Schwingungen. Zeitschrift für Flugwissenschaft und Weltraumforschung, vol. 12, pp. 358-367, 1988.

[25] Roberson R E., and Schwertassek R.: Dynamics of Multibody Systems Springer-Verlag, 1988

[26] Rosenberg R C., and Karnopp D.C.: Introduction to Physical System Dynamics McGraw-Hill, 1983

[27] Wehage R A.: Application of matrix partitioning and recursive projection to order n solution of constrained equations of motion In "Proc. of the 20th Biennial ASME Mechanisms Conf.", Orlando, Florida, 1988.

[28] Weiner M., and Cellier F E Modeling and Simulation of a Solar Energy System by Use of Bond Graphs Proceedings SCS Int Conf. on Bond Graph Modeling, San Diego, California, pp. 301–306 January 17-20, 1993

MINIMAL DYNAMIC CHARACTERIZATION OF TREE-LIKE MULTIBODY SYSTEMS

P. FISETTE, B. RAUCENT AND J.C. SAMIN
Université Catholique de Louvain
Département de Mécanique
2, Place du Levant B-1348 Louvain-La-Neuve, Belgium

ABSTRACT. The dynamic model of tree-like multibody systems is linear with respect to the parameters of mass distribution for instance when barycentric parameters are used. Thus, assuming that the parameters related to the kinematics are perfectly known, these quantities can be estimated through linear regression techniques. The necessary data are obtained by measuring the joint forces and/or torques and the resulting motion given in terms of positions, velocities and accelerations. An alternative method uses measurements of the reaction forces and torques applied to the bedplate.

The linearity of the dynamic and reaction models with respect to the barycentric quantities does not however imply that the latter constitute the minimum set of parameters characterizing the mass distribution of the system. In other words, some barycentric parameters may disappear from the models or may be redundant in the sense that they appear only via linear combinations. In the first case they are not identifiable, while in the second case the linear regression technique leads to estimated values which are correct for the combinations but can be erroneous for the individual parameters.

The various options taken to derive the dynamic and reaction models by use of the ROBOTRAN programme are briefly reviewed. Then the rules leading to the minimal parametrization are presented and illustrated by means of a practical example related to a robot calibration problem.

Introduction

Various problem in robotics require the computation of the dynamical model of the robot which relates the generalized control forces that are transmitted through the joints and the corresponding generalized coordinates q_i, their velocities and accelerations. For instance high speed and high precision control can only be achieved using advanced control algorithms, such as the "computed torque" control, which require an accurate and in-depth knowledge of the robot dynamics.

The structure itself of the dynamical equations is well known. However these equations contain parameters which correspond to the physical characteristics of the various bodies constituting the robot and therefore, a good knowledge of these parameters is absolutely necessary. In particular, measuring the parameters related to the mass distribution is not trivial and leads naturally to an identification problem [1,2,3,4].

In the first section of this paper, it is shown - often noted "a posterion" as a fact [4] - that the dynamical model is linear with respect to the parameters of mass distribution if

barycentric quantities are used [5]. However, it should be noted that this linearity with respect to these unknown parameters is valid only if the geometrical lengths of the system are supposed to be known, for instance by a kinematic calibration [6].

In order to emphasize these properties, the equations of motion of a tree-like multibody system have been written in a particular vectorial form derived from the Potential Power Principle (a modified version of d'Alembert Principle). As presented in section two, this form is used by the programme ROBOTRAN [7] to provide automatically the dynamical model. ROBOTRAN deals with mathematical expressions by means of pointers (PASCAL programming language) and prints the resulting equations in a literal form (character strings). One purpose of this programme is to emphasize the linearity with respect to barycentric parameters, so the derived equations are not necessarily optimal as regards the number of arithmetic operations for numerical simulations.

In the third section of this paper, an original procedure for the estimation of the barycentric parameters of a robot is presented [8, 9, 10]. This procedure is based on the property that the relations between the robot motion and its reactions on the bedplate are completely independent of the internal joints forces. The procedure thus requires only the processing of measurements provided by an external experimental set-up [8]. The robot is placed on a sensing platform which is provided with sensors able to measure the three components of the forces and the three components of the torques between the bedplate and the first link of the robot.

Section four deals with the parameter combinations which are required to satisfy the identifiability conditions. A systematic way to obtain these combinations is presented. In particular, it is shown that the reaction model used for identification has the same structure as the dynamical model and that all the barycentric parameters occuring in the dynamical model also appear explicitly in the identification model. Finally, a practical exemple of parameter combination is given in the fifth section.

1. Dynamic and Identification Models

1.1. GEOMETRICAL STRUCTURE AND KINEMATICS

In this section, we will review the various options taken to describe the multibody systems under consideration.

The system is considered as a set of n rigid bodies interconnected by joints. In the present paper, it is assumed that the structure is a topological tree. The joints are numbered in ascending order, starting from the base-plate (body 0) and each body has the same index as the preceding joint. A function, INBODY, is then defined to provide for every joint the index number of the preceding body. By means of a recurrent use of this function, the whole set of indices of bodies and joints of the kinematical chain which connect an arbitrary body to the base can be obtained. The notation $i \leq j$ means that the joint (body) i belongs to this chain for body j. For later convenience, a boolean matrix T is used whose elements are defined as :

$$T^{ij} = \begin{cases} 1 & \text{if } i \leq j \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

The characteristic geometrical dimensions of a body i (see fig 1) are given by the vectors $\underline{l}^{ij}$ and $\underline{l}^{ji}$. Two indices are needed because of the tree structure. However, although such

vectors should be defined only for consecutive bodies, the notation is extended to all the indices k satisfying $i < k$, defining $\underline{l}^{ik}$ as follows :

$$\underline{l}^{ik} = \underline{l}^{ij} \text{ for } \forall k \text{ such that } i < j \leq k \qquad (2)$$

Thus defined, $\underline{l}^{ik}$ provides the contribution of body i to the kinematical chain leading from the base to body k. Since the bodies are supposed to be rigid, all the vectors $\underline{l}^{ij}$ have constant components if they are expressed in a frame $(\hat{\underline{X}}^i)$ attached to body i.

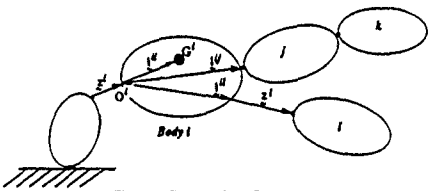


Fig. 1 : Geometrical Structure.

The joints are restricted to one degree of freedom : revolute or prismatic. There is no loss of generality since any physical joint can be modelled as a succession of such elementary joints. In order to develop the kinematics, we thus consider that each joint allows :

- a relative rotation expressed by the rotation matrix A^j that relates the j^{th} body frame $(\hat{\underline{X}}^j)$ to the previous one $(\hat{\underline{X}}^i)$. This matrix is constant for prismatic joints and depends on the angular coordinate q^j in the case of revolute joints.
- a relative translation represented by the vector $\underline{z}^j$. This vector is equal to zero for revolute joints and depends on the translational coordinate q^j in the case of prismatic joints.

The previous definitions are quite general and still allow us to choose the orientations of the body frames $(\hat{\underline{X}}^i)$ as well as the locations O^i of their origins (fig. 1) in order to minimize the number of geometric parameters such as for instance by using the Denavit and Hartenberg conventions. An equivalent minimal kinematic description (suggested in [11]) is illustrated in fig. 2 in the case of a serial link structure.

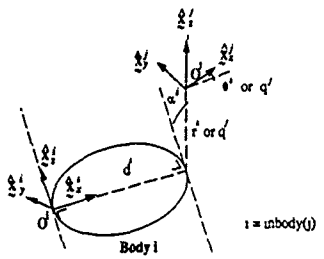


Fig. 2 : Kinematic Parameters [11].

1.2. MASS DISTRIBUTION PARAMETERS AND DYNAMICS

The dynamics of the system depend on the mass distribution of each body. The related parameters could be considered individually for each body, but some of these parameters would then combine to provide the actual parameters in the dynamic equations. These combinations could be performed manually [4] but this effort can be avoided if appropriate combinations are introduced from the outset by considering the position of the bodies in the topological structure [12].

Denoting by m^i the mass of the i^{th} body, by $\underline{l}^{ii}$ the position vector of its centre of mass G^i with respect to O^i (fig. 3) and by $\underline{I}^i$ its inertia tensor with respect to O^i, the following (barycentric) parameters are introduced :

$$\overline{m}^i = \sum_k T^{ik}\, m^k \tag{3}$$

$$\underline{mb}^i = \sum_k T^{ik}\, m^k\, \underline{l}^{ik} \tag{4}$$

and :

$$\underline{K}^i = \underline{I}^i - \sum_{k \neq i} T^{ik}\, m^k\, \underline{\tilde{l}}^{ik}\, \underline{\tilde{l}}^{ik} \tag{5}$$

where $\underline{\tilde{x}}$ denotes the skew-symmetric tensor associated with the cross product by a vector $\underline{x}$

These parameters are independent linear combinations of the basic mass parameters m^k, $m^k \underline{l}^{ik}$ and $\underline{I}^k$. Moreover, according to the previous definitions, $\overline{m}^i$ is constant and the moments of order 1 ($\underline{mb}^i$) and order 2 ($\underline{K}^i$) have constant components when expressed in the frame ($\hat{\underline{X}}^i$). These barycentric quantities may therefore be chosen as constant parameters which fully describe the mass distribution of the multibody system. Since there are 10 such scalar quantities for each body in the system, the total set of barycentric parameters is 10n.

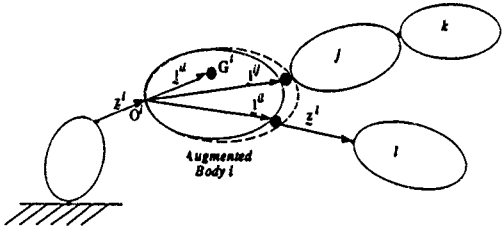


Fig. 3 : Augmented Body.

From a physical point of view, the mass $\overline{m}^i$ represents the mass of an augmented body i which consists of body i and point masses $\overline{m}^j$ (with i = INBODY(j)) located by the vector $\underline{l}^{ij}$.

The vector $\underline{mb}^i$ is the moment vector locating the mass centre of this augmented body and the tensor $\underline{K}^i$ is its inertia tensor with respect to its attachment point O^i.

As shown in [5] the Principle of Potential Power for rigid body systems leads to the following vector dynamic equations :

$$\underline{F}^j = \overline{m}^j \Big(\sum_{k:k<j} \big(\underline{\tilde{z}}^k + (\tilde{\underline{\omega}}^k + \tilde{\underline{\omega}}^k \tilde{\underline{\omega}}^k)\, \underline{l}^{kj} \big) - \underline{g} \Big) + \sum_{k} T^{jk} \big(\overline{m}^k \underline{z}^k + (\tilde{\underline{\omega}}^k + \tilde{\underline{\omega}}^k \tilde{\underline{\omega}}^k)\, \underline{mb}^k \big) \tag{6}$$

$$\underline{L}^j = \sum_{k} T^{jk} \big(\underline{z}^k \times \underline{F}^k + \underline{mb}^k \times (\underline{\tilde{z}}^k - \underline{g}) + \underline{K}^k \dot{\underline{\omega}}^k + \underline{\omega}^k \times \underline{K}^k \underline{\omega}^k \big) + \sum_{k} T^{jk} \sum_{l:l<k} \underline{mb}^k \times \big(\underline{\tilde{z}}^l + (\tilde{\underline{\omega}}^l + \tilde{\underline{\omega}}^l \tilde{\underline{\omega}}^l)\, \underline{l}^{lk} \big) + \sum_{k} T^{jk} \sum_{l:k<l} \underline{l}^{kl} \times \big(\overline{m}^l \underline{\tilde{z}}^l + (\tilde{\underline{\omega}}^l + \tilde{\underline{\omega}}^l \tilde{\underline{\omega}}^l)\, \underline{mb}^l \big) \tag{7}$$

and two important properties which are often noted "a posteriori" as a fact [4], arise :

i) *the torques $\underline{L}^j$ and the forces $\underline{F}^j$ are linear functions of the barycentric parameters $\overline{m}^k$, $\underline{mb}^k$, $\underline{K}^k$ with k such that $j \le k$. They become bilinear if the geometric lengths $\underline{l}^{jk}$ $(j<k)$ are also considered as parameters.*

ii) the definition of T^{jk} given in (1) allows us to observe that *$\underline{L}^i$ and $\underline{F}^i$, with i = INBODY(j), contains all the terms included in $\underline{L}^j$ and $\underline{F}^j$, and therefore the same barycentric parameters (in addition to those related to the augmented body i).*

In order to obtain the scalar equations of the dynamic model, we need to project the vectors $\underline{L}^j$ (or $\underline{F}^j$) onto the axis of its corresponding revolute (or prismatic) joint. Then all the products (scalar, vector, tensor) must be performed to obtain the final form of the dynamic model in which the generalized coordinates q^i, velocities $\dot{q}^i$ and accelerations $\ddot{q}^i$ appear explicitly. The latter can be written under the following matrix form :

$$M(q,\theta)\, \ddot{q} + F(q, \dot{q}, \theta) = Q \tag{8}$$

where : M is the (nxn) positive definite symmetric inertia matrix of the system, F is the n-vector specifying gravitation, Coriolis and centrifugal effects, Q is the n-vector of the generalized forces associated with q, and θ is the 10n-vector containing the barycentric parameters

Although the above operations preserve the linearity of the model with respect to the components of the barycentric parameters, the dynamic model involves a few independent combinations of those barycentric parameters. These combinations depend on the particular nature (either prismatic or revolute) of the joints and recursive methods for obtaining them

in a systematical way will be described in section 4. They lead to a set of N_d combined parameters denoted δ_d :

$$\delta_d = S_d \theta , \tag{9}$$

where S_d is a full rang matrix. Equation (23) can therefore be rewritten in the linear form

$$\phi_d (q, \dot{q}, \ddot{q}) \delta_d = Q , \tag{10}$$

where ϕ_d is a regressor vector depending on joint positions, velocities and accelerations

2. Description of the Robotran software

The main purpose of ROBOTRAN [7] is to provide a literal expression (character strings) of the dynamic model of any multibody system. By using the option dedicated to identification, the barycentric parameters appear explicitly in these expressions[1] so that the user can easily regroup their coefficients in order to obtain the regression vectors ϕ_d.

2.1. THE PROGRAMME SYNTAX

Each mathematical expression of equation (10) contains one or several terms linked by minus or plus signs, each term being the product of several factors. A factor can be an integer, a barycentric parameter, a generalized coordinate (q_i, $\dot{q}_i$, $\ddot{q}_i$), a geometrical constant, a trigonometrical function whose argument is a factor, or a mathematical expression between brackets. Thus defined, an expression can be considered as a tree whose nodes represent the nature of the expressions :

- integer whose value has to be given as a constant,
- barycentric parameter, generalized coordinate or geometrical constant which have to be identified by an appropriate string of characters,
- cosine or sine function whose argument (also an expression) must be given,
- plus, minus, times operators whose two operands must be given as expressions.

For instance, the expression $(g * \sin(q_1) + a * \ddot{q}_2)$ can be represented by fig 4 :

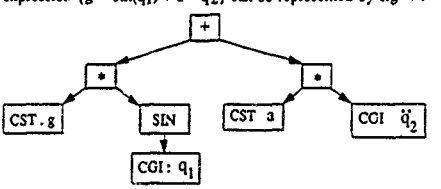


Fig. 4 : Tree representation of a symbolical expression.

[1] These expressions are thus not necessarily optimal as regards the number of arithmetic operations for on-line applications.

2.2. THE MANIPULATIONS AND SIMPLIFICATIONS

In order to avoid expressions like ' x = a + b - a, some order is introduced to classify the expressions. This order is given first by their *natures* and then by a lexicographical order on the strings of characters (for instance, $0 < \cos(q_1) < \sin(q_1) < (q_2+q_3)$). At each operation, the resulting expressions are rewritten according to the prescribed order and then consecutive equal terms with opposite signs are simplified. It can thus be ensured that the expression appearing in the obtained equations cannot be further reduced without using specific formulae of trigonometry. This kind of reductions can be performed manually or by several dedicated programmes (MACSYMA, SMP, REDUCE, ...).

Finally, in order to reduce the size of the output, intermediate auxiliary variables can be used to replace each product of trigonometrical functions. These variables are produced by taking the relation order into account so it is as easy as before to find the possible combinations of barycentric parameters and the regression vectors.

2.3. APPLICATION TO THE PUMA ROBOT

The PUMA 562 (UNIMATION®) is a serial six degree of freedom manipulator with revolute joints (fig. 5). Assuming here that the influence of the wrist is negligible, only the first three joints are considered.

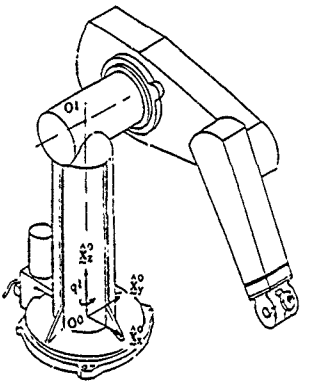


Fig. 5 : The PUMA 562 (UNIMATION®).

Three body-fixed frames have been introduced for the dynamic modelling. In the reference configuration, when the wrist is located straight above the shoulder, all these frames are aligned with the inertial frame $\{X^0\}$ attached to the bedplate. The physical characteristics (geometry and mass distribution) of each body are given in table 1.

	mass	Central Inertia Tensor			Center of mass position	Joint position
link 1	m1	J^1_{xx}	0	0	0	0
		0	J^1_{yy}	0	l^1_{cy}	0
		0	0	J^1_{zz}	l^1_{cz}	0
link 2	m2	J^2_{xx}	0	0	0	0
		0	J^2_{yy}	0	l^2_{cx}	0
		0	0	J^2_{zz}	l^2_{cz}	l^2_2
link 3	m3	J^3_{xx}	0	0	0	0
		0	J^3_{yy}	0	l^3_{cx}	0
		0	0	J^3_{zz}	l^3_{cy}	l^3_{cz}

Table 1 : PUMA Description

According to the symbols defined in this table, the barycentric parameters and the dynamic model are obtained automatically by the software ROBOTRAN :

Barycentric masses

mm1 = m1+m2+m3
mm2 = m2+m3
mm3 = m3

Barycentric moments[2]

mb1y = l11y*m1
mb1z = mm2*l12z+l11z*m1
mb2y = l22y*m2
mb2z = mm3*l23z+l22z*m2
mb3y = l33y*m3
mb3z = l33z*m3

Barycentric tensors

K1xx = J1xx+mm2*l12z*l12z-m1*(-l11y*l11y-l11z*l11z)
K1yy = J1yy+mm2*l12z*l12z+l11z*l11z*m1
K1yz = -l11y*l11z*m1
K1zz = J1zz+l11y*l11y*m1
K2xx = J2xx+mm3*l23z*l23z-m2*(-l22y*l22y-l22z*l22z)
K2yy = J2yy+mm3*l23z*l23z+l22z*l22z*m2
K2yz = -l22y*l22z*m2
K2zz = J2zz+l22y*l22y*m2
K3xx = J3xx-m3*(-l33y*l33y-l33z*l33z)
K3yy = J3yy+l33z*l33z*m3
K3yz = -l33y*l33z*m3
K3zz = J3zz+l33y*l33y*m3

Equations[3]

Q3 = K3yz*qpp1*C23
+K3yy*qpp2
+mb3z*qpp2*l23z*C3
+K3yy*qpp3
-K3xx*qp1*qp1*C23*S23
+K3zz*qp1*qp1*C23*S23
-mb3z*qp1*qp1*l23z*C23*S2
+mb3z*qp2*qp2*l23z*S3
mb3z*g*S23

Q2 = Q3
+K2yz*qpp1*C2
-mb3y*qpp1*l2? *C2
+K2yy*qpp2
+mb3z*qpp2*l23z*C3
+mb3z*qpp3*l23z*C3
-K2xx*qp1*qp1*C2*S2
+K2zz*qp1*qp1*C2*S2
-mb3z*qp1*qp1*l23z*C2*S23
-mb3z*qp2*qp2*l23z*S3
-mb3z*qp3*qp3*l23z*S3
-mb3z*2*qp2*qp3*l23z*S3
-mb2z*g*S2

[2] The particular values given in table 1 lead to a zero value for mbix, Kixy and Kixz (with i= 1, 2 and 3)

[3] where for instance, qp1 stands for $\dot{q}(1)$, qpp1 for $\ddot{q}(1)$, S2 for SIN(q(2)), C2 for COS(q(2)), S223 for SIN(q(2)+q(2)+q(3))...

Q1 = K1zz*qpp1
+K2xx*qpp1*S2*S2
+K2zz*qpp1*C2*C2
+K3xx*qpp1*S23*S23
+K3zz*qpp1*C23*C23
+mb3z*2*qpp1*l23z*S2*S23
+K2yz*qpp2*C2
+K3yz*qpp2*C23
-mb3y*qpp2*l23z*C2
+K3yz*qpp3*C23

-K2yz*qp2*qp2*S2
-K3yz*qp2*qp2*S23
+mb3y*qp2*qp2*l23z*S2
+K2xx*qp1*qp2*S22
-K2zz*qp1*qp2*S22
+K3xx*qp1*qp2*S2233
-K3zz*qp1*qp2*S2233
+mb3z*2*qp1*qp2*l23z*S223
-K3yz*qp3*qp3*S23
+K3xx*qp1*qp3*S2233
-K3zz*qp1*qp3*S2233
+mb3z*2*qp1*qp3*l23z*C23*S2
-K3yz*2*qp2*qp3*S23

3. Identification of Dynamic Parameters

In the classical identification approach the values of δ_d are estimated from input data (torques applied to the links) and output data (positions, velocities and accelerations of the joints) provided by *"internal" measurement* devices located inside the arms (see e.g.[1,2,3]). The model relating these input and output variables is described by the linear equations (10). Therefore, in principle, parameters can be estimated through linear regression techniques. However, there is a major drawback in the practical implementation of such techniques : direct measurements of torques applied to the links are not available, so that torques have to be evaluated as sums of torques provided by the actuators and of friction torques which may be relatively large. Two problems then occur :

a) For most commercial robots, torques provided by the actuators can be obtained from *internal measurements*, but with poor accuracy. Consider for instance a permanent magnet d-c motor controlled through its armature voltage or current. The torque can be estimated from input current measurements using the torque constant available from the manufacturer's technical data, albeit with uncertainties up to 10% ; furthermore the value of this constant can change over the robot lifetime.

b) The implementation of a parameter estimation procedure requires an accurate model of friction effects and estimation of the characteristic parameters of the friction. It means that barycentric parameters and friction parameters must be estimated simultaneously, based on an adequate friction model. This coupling between barycentric and friction parameters may degrade the accuracy of the barycentric parameter estimates.

An alternative approach has been proposed in order to avoid the above drawbacks [8,9,10]. It is based on a reformulation of the system dynamics relating the motion of the robot to the reaction forces and torques on its bedplate. The robot is placed on a sensing platform equipped with sensors providing measurements of the three forces and the three torques components between the bedplate and the robot. The main advantage of such an experimental set-up is that it supplies the estimation algorithm with data which are far more accurate than the data which could be obtained from classical sensors located inside the robot arms. Furthermore, the reaction model is *independent from internal effects* (i.e. friction)

Analytical expressions of the reaction model can be obtained for the reaction torque and force components on the bedplate by projecting the vectors L^1 and F^1 (given by Eqs. (6) and (7)) onto the axes of the inertial reference frame attached to the bedplate. As a result of conclusion (1.2.ii), all the barycentric parameters contained in the vectorial joint equations (6) and (7) will appear in L^1 and F^1, and therefore in the reaction model which contains all the components of these two vectors. Moreover, the few linear combinations δ_r of barycentric parameters which will appear when projecting L^1 and F^1 on the axes of the bedplate frame, contain the set of linear combination δ_d which occur while deducing the dynamical model (8) from its vectorial form (6-7). In other words, the combined barycentric parameters set δ_d defined in (9) is a subset of the N_r parameters δ_r which appear independently in the reaction model :

$$\delta_d = S\,\delta_r \quad \text{with } \delta_r = S_r\,\theta \tag{11}$$

The reaction model has thus the form :

$$\phi_r(q, \dot{q}, \ddot{q})\,\delta_r = Q_r, \tag{12}$$

where ϕ_r is a regressor vector depending on joint positions, velocities and accelerations and where the Q_r vector contains the six components of force/torque reactions between the bedplate and the robot.

Assuming that the geometric parameters are perfectly known from a static calibration [6], the barycentric parameters can be estimated through linear regression techniques based on the dynamic or reaction model. Indeed by taking measurements of Q(t), q(t), $\dot{q}(t)$ and $\ddot{q}(t)$ for different values $t_1, t_2, \ldots, t_M$ of time, the following linear regression model is obtained either from (10) or (12) :

$$\Phi\,\delta = Z \tag{13}$$

where : $Z = \left(Q^T(t_1), \ldots, Q^T(t_M)\right)^T$ and

$$\Phi = \left(\phi(q(t_1), \dot{q}(t_1), \ddot{q}(t_1)), \ldots, \phi(q(t_M), \dot{q}(t_M), \ddot{q}(t_M))\right)^T.$$

The problem of the identifiability of δ is stated as follows : does there exist only one set of values of δ satisfying the linear regression model (13) for any motion ? We have the following trivial result :

"δ is identifiable iff there exists a trajectory of the robot such that Φ has full rank".

Indeed in this case :

$$\delta = \left(\Phi^T\Phi\right)^{-1}\Phi^T Z \tag{14}$$

In order to satisfy this condition, we have to avoid redundant parametrizations and to select sufficiently rich trajectories.

(a) to avoid redundant parametrizations of the model [15,16,17]. A parametrization δ is said to be redundant for the model, if there exists another parametrization δ^* of dimension N^* with $N^* < N$, a full rank matrix S and a regressor Φ^* such that along any trajectory :

$$\delta^* = S^*\,\delta \quad \text{and} \quad Z = \Phi\,\delta = \Phi^*\,\delta^* \tag{15}$$

In such a case it is clear that Φ cannot be full rank. For most mechanical structures, the full set of barycentric parameters is redundant for the dynamic and reaction models

(b) to select trajectories which are sufficiently rich In [13] Gauthier describes a procedure for the automatic generation of test trajectories which guarantee the identifiability rank conditions of Φ assuming a non-redundant parametrization. We have however observed in practical experiments that an empirical selection of such trajectories is fairly easy when condition (a) is satisfied [14].

4. Minimal Dynamic Parametrization

4.1. THE RECURSIVE MODEL

In this section a recursive method for obtaining the minimal set of parameters δ in a systematic way will be developed. Since any joint mobility is restricted to one degree of freedom (i.e. revolute or prismatic), the two models depicted in fig. 6 and fig. 7 will be considered. These systems are composed of a body j and a carrying body i (with i=INBODY(j)). Depending on the kinematic chain which links body i to the base, this carrying body may have up to six degrees of freedom with respect to inertial space.

According to section 1, frames $\{\underset{\sim}{\hat{X}}^i\}$ and $\{\underset{\sim}{\hat{X}}^j\}$ are attached to the bodies and the axis vector of the joint j is denoted by $\underset{\sim}{e}^j$.

4.2. PRISMATIC JOINT

The angular velocity of the carrying body i is represented by $\underset{\sim}{\omega}^i$ and the absolute position of O^i is given by the vector $\underset{\sim}{x}^i$:

$$\underset{\sim}{x}^i = \sum_{k:k<i} (\underset{\sim}{z}^k + \underset{\sim}{l}^{ki}) + \underset{\sim}{z}^i \qquad (16)$$

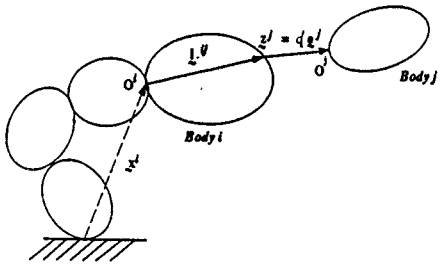


Fig. 6 : Prismatic joint recursive model.

Since no a priori assumption is made concerning the mobility of the carrying body 1, the vector forms of eqs. (6) and (7) are retained for the body i :

$$\underline{F}^i = \overline{m}^i(\ddot{\underline{x}}^i - \underline{g}) + (\dot{\underline{\tilde{\omega}}}^i + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\,\underline{mb}^i + \overline{m}^j\,\ddot{\underline{z}}^j + (\dot{\underline{\tilde{\omega}}}^j + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\underline{mb}^j \tag{17}$$

$$\begin{aligned}\underline{L}^i = {} & \underline{z}^i \times \underline{F}^i + \underline{mb}^i \times (\ddot{\underline{x}}^i - \underline{g}) + \underline{K}^i\,\dot{\underline{\omega}}^i + \underline{\omega}^i \times \underline{K}^i\,\underline{\omega}^i \\ & + \underline{I}^{ij} \times (\overline{m}^j\,\ddot{\underline{z}}^j + (\dot{\underline{\tilde{\omega}}}^j + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\underline{mb}^j) \\ & + \underline{z}^j \times \underline{F}^j + \underline{mb}^j \times (\ddot{\underline{z}}^j - \underline{g}) + \underline{K}^j\,\dot{\underline{\omega}}^j + \underline{\omega}^j \times \underline{K}^j\underline{\omega}^j \\ & + \underline{mb}^j \times (\ddot{\underline{x}}^i + (\dot{\underline{\tilde{\omega}}}^i + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\underline{I}^{ij})\end{aligned} \tag{18}$$

where : $$\underline{F}^j = \overline{m}^j(\ddot{\underline{x}}^j - \underline{g}) + (\dot{\underline{\tilde{\omega}}}^j + \underline{\tilde{\omega}}^j\,\underline{\tilde{\omega}}^j)\underline{mb}^j \tag{19}$$

and the dynamic equation of joint j is given by :

$$F^j = \underline{F}^j \cdot \underline{e}^j \tag{20}$$

As a preliminary conclusions of property (1.2.II), the equations (17) and (18) contain the contribution of all the barycentric parameters related to the two considered bodies. For a general system in which the j^{th} body is not a terminal one, these equations would involve additional terms corresponding to the barycentric quantities associated with the bodies located downstream from body j. Equations (17), (18) and (20) can therefore be used to detect, at each step of a recursive reasoning, the combinations which may occur in the models.

The various barycentric parameters of body j have now to be investigated in order to establish the recursive relations leading to δj, the minimal parametrization related to body j.

Since the frame $\{\hat{\underline{X}}^j\}$ has no relative angular motion with respect to $\{\hat{\underline{X}}^i\}$, the components of $\underline{mb}^j$ and $\underline{K}^j$ are also constant in the $\{\hat{\underline{X}}^i\}$ frame. Using the following properties :

$$\underline{\tilde{b}}\,\underline{\tilde{\omega}}\,\underline{\tilde{\omega}}\,\underline{\tilde{a}} + \underline{\tilde{a}}\,\underline{\tilde{\omega}}\,\underline{\tilde{\omega}}\,\underline{\tilde{b}} = -(\underline{\tilde{b}}\,\underline{\tilde{a}} + \underline{\tilde{a}}\,\underline{\tilde{b}})\,\dot{\underline{\omega}} \tag{21}$$

$$\underline{\tilde{b}}\,\underline{\tilde{\omega}}\,\underline{\tilde{\omega}}\,\underline{\tilde{a}} + \underline{\tilde{a}}\,\underline{\tilde{\omega}}\,\underline{\tilde{\omega}}\,\underline{\tilde{b}} = -\underline{\tilde{\omega}}\,(\underline{\tilde{b}}\,\underline{\tilde{a}} + \underline{\tilde{a}}\,\underline{\tilde{b}})\,\underline{\omega} \tag{22}$$

we can write : $$\underline{I}^{ij} \times (\dot{\underline{\tilde{\omega}}}^i + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\underline{mb}^j + \underline{mb}^j \times (\dot{\underline{\tilde{\omega}}}^i + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\underline{I}^{ij}$$

$$= -(\underline{\tilde{I}}^{ij}\underline{\tilde{mb}}^j + \underline{\tilde{mb}}^j\underline{\tilde{I}}^{ij})\dot{\underline{\omega}}^i - \underline{\tilde{\omega}}^i(\underline{\tilde{I}}^{ij}\underline{\tilde{mb}}^j + \underline{\tilde{mb}}^j\underline{\tilde{I}}^{ij})\underline{\omega}^i$$

and looking at eq. (18), we may suggest the following combinations :

$$\begin{aligned}\underline{mb}^{*i} &= \underline{mb}^i + \underline{mb}^j \\ \underline{K}^{*i} &= \underline{K}^i + \underline{K}^j - (\underline{\tilde{I}}^{ij}\underline{\tilde{mb}}^j + \underline{\tilde{mb}}^j\underline{\tilde{I}}^{ij})\end{aligned} \tag{23}$$

where it must be noted that all vectors and tensors have constant components in the $\{\hat{\underline{X}}^i\}$ frame.

Using (23), equations (17), (18) and (20) are now rewritten as :

$$\underline{F}^i = \overline{m}^i(\ddot{\underline{x}}^i - \underline{g}) + (\tilde{\dot{\omega}}^i + \tilde{\omega}^i\tilde{\omega}^i)\,\underline{mb}^{*i} + \overline{m}^j\ddot{\underline{z}}^j \tag{24}$$

$$\underline{L}^i = \underline{z}^i \times \underline{F}^i + \underline{mb}^{*i} \times (\ddot{\underline{x}}^i - \underline{g}) + \underline{K}^{*i}\dot{\underline{\omega}}^i + \underline{\omega}^i \times \underline{K}^{*i}\underline{\omega}^i + \underline{l}^{ij} \times \overline{m}^j\ddot{\underline{z}}^j + \underline{z}^j \times \underline{F}^j + \underline{mb}^j \times \ddot{\underline{z}}^j \tag{25}$$

$$F^j = \underline{e}^j \cdot \overline{m}^j(\ddot{\underline{x}}^j - \underline{g}) + \underline{e}^j \cdot (\tilde{\dot{\omega}}^i + \tilde{\omega}^i\tilde{\omega}^i)\underline{mb}^j \tag{26}$$

This new presentation clearly shows that $\underline{K}^j$ is redundant. On the contrary $\overline{m}^j$ can easily be identified from (26) by choosing an excitation such that $\ddot{\underline{x}}^j = \underline{e}^j\ddot{q}^j$. Finally, the identifiability of $\underline{mb}^j$ can be analysed by means of its individual contribution to $\underline{L}^i$ given by

$$\underline{e}^j q^j(\tilde{\dot{\omega}}^i + \tilde{\omega}^i\tilde{\omega}^i)\,\underline{mb}^j + \underline{mb}^j \times (\underline{e}^j\ddot{q}^j + 2\,\tilde{\omega}^i\underline{e}^j\dot{q}^j + (\tilde{\dot{\omega}}^i + \tilde{\omega}^i\tilde{\omega}^i)\underline{e}^j q^j) \tag{27}$$

and by its contribution to F^j given by :

$$\underline{e}^j \cdot (\tilde{\dot{\omega}}^i + \tilde{\omega}^i\tilde{\omega}^i)\underline{mb}^j \tag{28}$$

This analysis leads to different combinations depending on the particular values of $\underline{\omega}^i$, [17]. In the general case, expressions (27) and (28) allow the identification of the three components of $\underline{mb}^j$ using the dynamic or the reaction model. It is therefore unnecessary to include the barycentric vector $\underline{mb}^j$ in the parameter combinations suggested in (23).

If the mobility of body i is restricted (i.e. $\underline{\omega}^i$ is inertially fixed), the minimal set of parameters of the dynamic model depends on expressions (28) and the projection of (27) onto the fixed direction $\underline{w}$ of $\underline{\omega}^i$. In this case the component $mb^j_{\underline{w}}$ of $\underline{mb}^j$ along this direction is not identifiable using the dynamic model. Moreover, if $\underline{\omega}^i = \underline{0}$ or if the direction of $\underline{\omega}^i$ coincides with $\underline{e}^j$, the two normal components $mb^j_{\underline{u}}$ and $mb^j_{\underline{v}}$ also disappear from (28) and the projection of (27). In both cases the parameter combinations (23) must consequently be applied to the components of $\underline{mb}^j$ which become redundant for the dynamic model. The results may be summarized as indicated in table 2 for the dynamic model.

$\underline{\omega}^i$ *angular velocit of body i*	δ_d^j *minimal set of body j parameters*	θ_d^{*i} *redefinition of the carrying body i barycentric parameters*
general case	$(\overline{m}^j, mb_x^j, mb_y^j, mb_z^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j$
$\underline{\omega}^i$ inertially fixed : $\underline{\omega}^i = \omega^i \ \underline{w}$	$(\overline{m}^j, mb_u^j, mb_v^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_w^j \ \underline{w}$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j - mb_w^j \ (\underline{\tilde{I}}^{ij} \ \underline{\tilde{w}} + \underline{\tilde{w}} \ \underline{\tilde{I}}^{ij})$
$\underline{\omega}^i$ inertially fixed with $\underline{\omega}^j = \omega^i \ \underline{e}^j$ or $\underline{\omega}^i = \underline{0}$	$(\overline{m}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + \underline{mb}^j$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j - (\underline{\tilde{I}}^{ij} \ \underline{\tilde{mb}}^j + \underline{\tilde{mb}}^j \ \underline{\tilde{I}}^{ij})$

Table 2 . Combination rules for prismatic joints in the dynamic model.

Similar considerations can be applied to the reaction model in which less components are redundant because this model can retain all the components of relation (27) The results are summarized as indicated in table 3.

$\underline{\omega}^i$ *angular velocit of body i*	δ_r^j *minimal set of body j parameters*	θ_r^{*i} *redefinition of the carrying body i barycentric parameters*
general case	$(\overline{m}^j, mb_x^j, mb_y^j, mb_z^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j$
$\underline{\omega}^i$ inertially fixed : $\underline{\omega}^i = \omega^i \ \underline{w}$	$(\overline{m}^j, mb_u^j, mb_v^j, mb_w^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j$
$\underline{\omega}^i$ inertially fixed with $\underline{\omega}^j = \omega^i \ \underline{e}^j$ or $\underline{\omega}^i = \underline{0}$	$(\overline{m}^j, mb_x^j, mb_y^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \ \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i + \underline{K}^j - mb_z^j \ (\underline{\tilde{I}}^{ij} \ \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \ \underline{\tilde{I}}^{ij})$

Table 3 . Combination rules for prismatic joints in the reaction model.

For both models, the indicated rules must be applied recursively from the end(s) of the structure towards its base in order to determine the minimal sets of parameters δ^j for the whole system

4.3 REVOLUTE JOINT

The angular velocity and acceleration of body j are :

$$\underline{\omega}^j = \underline{\omega}^i + \underline{e}^j \dot{q}^j \tag{29}$$

$$\underline{\dot{\omega}}^j = \underline{\dot{\omega}}^i + \underline{\omega}^i \times \underline{e}^j \dot{q}^j + \underline{e}^j \ddot{q}^j \tag{30}$$

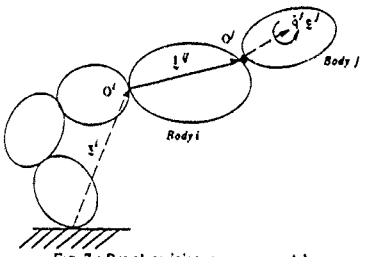


Fig. 7 : Revolute joint recursive model.

Since again no a priori assumption is made concerning the mobility of the carrying body i for the general case, the vector forms of eqs. (6) and (7) are retained :

$$\underline{F}^i = \overline{m}^i(\underline{\ddot{X}}^i - \underline{g}) + (\underline{\tilde{\dot{\omega}}}^i + \underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i)\,\underline{mb}^i + \overline{m}^j\,\underline{\ddot{z}}^j + (\underline{\tilde{\dot{\omega}}}^j + \underline{\tilde{\omega}}^j\,\underline{\tilde{\omega}}^j)\underline{mb}^j \tag{31}$$

$$\underline{L}^i = \underline{z}^i \times \underline{F}^i + \underline{mb}^i \times (\underline{\ddot{X}}^i - \underline{g}) + \underline{K}^i\,\underline{\dot{\omega}}^i + \underline{\omega}^i \times \underline{K}^i\,\underline{\omega}^i + \underline{l}^{ij} \times (\underline{\tilde{\dot{\omega}}}^j + \underline{\tilde{\omega}}^j\,\underline{\tilde{\omega}}^j)\underline{mb}^j + \underline{L}^j \tag{32}$$

where

$$\underline{L}^j = \underline{K}^j\,\underline{\dot{\omega}}^j + \underline{\omega}^j \times \underline{K}^j\,\underline{\omega}^j + \underline{mb}^j \times (\underline{\ddot{X}}^j - \underline{g}) \tag{33}$$

and the dynamic equation of joint j can be written :

$$D^j = \underline{L}^j \cdot \underline{e}^j \tag{34}$$

The barycentric mass $\overline{m}^j$ does not appear anywhere in the equations and is therefore redundant. Using eqs (29), (30) and the following relation :

$$\underline{\tilde{\dot{\omega}}}^j\,\underline{\tilde{\omega}}^j\,\underline{\tilde{\omega}}^j = \underline{\tilde{\dot{\omega}}}^i\,\underline{\tilde{\omega}}^i\,\underline{\tilde{\omega}}^i + 2\,\underline{\tilde{\omega}}^i\,\underline{\tilde{e}}^j\,\dot{q}^j + \underline{\tilde{e}}^j\,\ddot{q}^j + \underline{\tilde{e}}^j\,\underline{\tilde{e}}^j\,(\dot{q}^j)^2 \tag{35}$$

the properties (21) and (22) lead to the possible combination (23) defined in the case of a prismatic joint. However, in the present case, $\{\hat{X}^j\}$ has a relative angular motion with respect to $\{\hat{X}^i\}$. The only component of $\underline{mb}^j$ that remains constant in the frame $\{\hat{X}^i\}$ is mb_z^j. In order to determine which components of $\underline{K}^j$ are concerned, it is decomposed in an auxiliary frame

$\hat{\underline{Y}}^{(j)}$ attached to body j. Without loss of generality, this frame is chosen such that $\hat{\underline{Y}}^j_z = \underline{e}^j$

Due to the rotation φ, the components of $\underline{K}^j$ in the $\{\hat{\underline{Y}}^{(j)}\}$ basis are given by means of the symmetric matrix

$$\begin{pmatrix} \cos(2\varphi)K^j_d - \sin(2\varphi)K^j_{xy} + K^j_s & \cdots & \\ \cos(2\varphi)K^j_{xy} + \sin(2\varphi)K^j_d & -\cos(2\varphi)K^j_d + \sin(2\varphi)K^j_{xy} - K^j_s & \\ \cos(\varphi)K^j_{xz} - \sin(\varphi)K^j_{yz} & \sin(\varphi)K^j_{xz} + \cos(\varphi)K^j_{yz} & K^j_{zz} \end{pmatrix} \tag{36}$$

where : $$K^j_s = \frac{K^j_{xx} + K^j_{yy}}{2} \quad \text{and} \quad K^j_d = \frac{K^j_{xx} - K^j_{yy}}{2} \tag{37}$$

As a preliminary result, this expression shows that the components K^j_{xx} and K^j_{yy} will only appear through the combinations K^j_s and K^j_d. The component K^j_{zz} is obviously identifiable in $\underline{L}^j \cdot \underline{e}^j$ via an excitation motion for which $\underline{\omega}^j = \underline{e}^j \dot{\varphi}^j$. The following combinations

$$\boxed{\begin{aligned} \underline{mb}^{*j} &= \underline{mb}^j + mb^j_z \underline{e}^j \\ \underline{K}^{*j} &= \underline{K}^j - mb^j_z \left(\underline{\tilde{l}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{l}}^{ij} \right) - K^j_s \underline{\tilde{e}}^j \underline{\tilde{e}}^j \end{aligned}} \tag{38}$$

are still applicable. Indeed it is easy to see that if we introduce these combinations into equation (31) of $\underline{F}^i$, then use (35) the only residual term that involves mb^j_z independently is

$$\left(2\, \underline{\tilde{\omega}}^j \underline{\tilde{e}}^j \dot{\varphi}^j + \underline{\tilde{e}}^j \underline{\tilde{e}}^j \ddot{\varphi}^j + \underline{\tilde{e}}^j \underline{\tilde{e}}^j (\dot{\varphi}^j)^2 \right) \underline{e}^j\, mb^j_z \tag{39}$$

and vanishes because $\underline{\tilde{e}}^j \underline{e}^j = \underline{0}$. Introducing (38) into equation (32) of $\underline{L}^i$ leads to the following residual term :

$$\left(\underline{z}^i + \underline{l}^{ij} \right) \times \left(2\, \underline{\tilde{\omega}}^j \underline{\tilde{e}}^j \dot{\varphi}^j + \underline{\tilde{e}}^j \ddot{\varphi}^j + \underline{\tilde{e}}^j \underline{\tilde{e}}^j (\dot{\varphi}^j)^2 \right) \underline{e}^j\, mb^j_z \tag{40}$$

which vanishes for the same reason. The individual contribution of mb^j_z also disappears in (34). As a consequence, this component cannot be identified and is redundant. Similarly, using (29) and (30), the individual contribution of $\underline{K}^j_s = - K^j_s \underline{\tilde{e}}^j \underline{\tilde{e}}^j$ reduces in (32) to

$$\underline{K}^j_s \underline{e}^j \ddot{\varphi}^j + \underline{\tilde{e}}^j \dot{\varphi}^j \underline{K}^j_s \underline{e}^j \dot{\varphi}^j + \left(\underline{\tilde{e}}^j \underline{K}^j_s - \underline{K}^j_s \underline{\tilde{e}}^j - (\underline{K}^j_s \underline{e}^j)^{\sim} \right) \underline{\omega}^j \dot{\varphi}^j \tag{41}$$

and in (34) to :

$$\underline{e}^j \cdot \underline{K}^j_s \dot{\underline{\omega}}^j + \underline{e}^j \cdot \underline{\tilde{\omega}}^j \underline{K}^j_s \underline{\omega}^j \tag{42}$$

It is easy to verify that both residual contributions vanish so that K^j_s is also a redundant parameter.

In the general case, i e when the mobility of the carrying body is not restricted, all the other components of $\underline{mb}^j$ and $\underline{K}^j$ are identifiable in the dynamic model as well as in the reaction model [10,17]. As for the prismatic joint, if $\underline{\omega}^i$ is inertially fixed additional components of $\underline{mb}^j$ and $\underline{K}^j$ can disappear. The results are summarized in tables 4 and 5 respectively for the dynamic and reaction model.

$\underline{\omega}^i$ *angular velocity of the carrying body i*	δ_d^j *minimal set of body j parameters*	θ_d^{*i} *redefinition of the carrying body i barycentric parameters*
general case	$(mb_x^j, mb_y^j, K_{xy}^j, K_{xz}^j, K_{yz}^j, K_d^j, K_{zz}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$
$\underline{\omega}^i = \omega^i \underline{e}^j$	$(mb_x^j, mb_y^j, K_{zz}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$
$\underline{\omega}^i = \omega^i \underline{e}^j$, $\underline{\ddot{x}}^j = \ddot{x}^j \underline{e}^j$, and $\underline{g} = g \underline{e}^j$.	(K_{zz}^j)	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$

Table 4 : Combination rules for revolute joints in the dynamic model.

$\underline{\omega}^i$ *angular velocity of the carrying body i*	δ_r^j *minimal set of body j parameters*	θ_r^{*i} *redefinition of the carrying body i barycentric parameters*
general case	$(mb_x^j, mb_y^j, K_{xy}^j, K_{xz}^j, K_{yz}^j, K_d^j, K_{zz}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$
$\underline{\omega}^i = \omega^i \underline{e}^j$	$(mb_x^j, mb_y^j, K_{xz}^j, K_{yz}^j, K_{zz}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$
$\underline{\omega}^i = \omega^i \underline{e}^j$, $\underline{\ddot{x}}^j = \ddot{x}^j \underline{e}^j$, and $\underline{g} = g \underline{e}^j$	$(mb_x^j, mb_y^j, K_{xz}^j, K_{yz}^j, K_{zz}^j)$	$\overline{m}^{*i} = \overline{m}^i$ $\underline{mb}^{*i} = \underline{mb}^i + mb_z^j \underline{e}^j$ $\underline{K}^{*i} = \underline{K}^i - mb_z^j (\underline{\tilde{I}}^{ij} \underline{\tilde{e}}^j + \underline{\tilde{e}}^j \underline{\tilde{I}}^{ij}) - K_z^j \underline{\tilde{e}}^j \underline{\tilde{e}}^j$

Table 5 : Combination rules for revolute joints in the reaction model.

5. Application to the PUMA robot

According to the recursive rules given in table 5, the redefined (non-zero) barycentric parameters θ^* of the PUMA robot are :

$$\theta_d^* = \begin{pmatrix} \overline{m}^{*3} & mb_y^{*3} & mb_z^{*3} & K_{yy}^{*3} & K_d^{*3} & K_s^{*3} & K_{yz}^{*3} \\ \overline{m}^{*2} & mb_y^{*2} & mb_z^{*2} & K_{yy}^{*2} & K_d^{*2} & K_s^{*2} & K_{yz}^{*2} \\ \overline{m}^{*1} & mb_y^{*1} & mb_z^{*1} & K_{zz}^{*1} & K_d^{*1} & K_s^{*1} & K_{yz}^{*1} \end{pmatrix}$$

among which the minimal set of non redundant parameters is given by :

$$\delta_d = \begin{pmatrix} mb_z^3 & K_{yy}^3 & K_d^3 & K_{yz}^3 \\ mb_y^{*2} & K_{yy}^{*2} & K_d^{*2} & K_{yz}^{*2} \\ & & & K_{zz}^{*1} \end{pmatrix}$$

Indeed, the barycentric masses $\overline{m}^{i}$ (i=1,....,3) are redundant while mb_y^{*1}, K_d^{*1} and K_{yz}^{*1} disappear because of the restricted mobility of body 1 ($\underline{\omega}^0 = \underline{0}$, $\underline{\dot{v}}^1 = \underline{0}$, and $\underline{g} = g\,\underline{e}^1$).

For example the equation Q1 of the first joint can be rewritten as follows :

OUTPUT ROBOTRAN

```
Q1 = K1zz*qpp1
   +K2xx*qpp1*S2*S2
   +K2zz*qpp1*C2*C2
   +K3xx*qpp1*S23*S23
   +K3zz*qpp1*C23*C23
   +mb3z*2*qpp1*l23z*S2*S23
   +K2yz*qpp2*C2
   +K3yz*qpp2*C23
   -mb3y*qpp2*l23z*C2
   +K3yz*qpp3*C23
   -K2yz*qp2*qp2*S2
   -K3yz*qp2*qp2*S23
   +mb3y*qp2*qp2*l23z*S2
   +K2xx*qp1*qp2*S22
    K2zz*qp1*qp2*S22
   +K3xx*qp1*qp2*S2233
   -K3zz*qp1*qp2*S2233
   +mb3z*2*qp1*qp2*l23z*S223
   -K3yz*qp3*qp3*S23
   +K3xx*qp1*qp3*S2233
   -K3zz*qp1*qp3*S2233
   +mb3z*2*qp1*qp3*l23z*C23*S2
   -K3yz*2*qp2*qp3*S23
```

REDUCED DYNAMIC EQUATION

```
Q1 = K*1zz*qpp1
   +K*2d*qpp1*C22
   +K3d*qpp1*C2233
   +mb3z*2*qpp1*l23z*S2*S23
   +K*2yz*qpp2*C2
   +K3yz*qpp2*C23
   +K3yz*qpp3*C23
   -K*2yz*qp2*qp2*S2
   -K3yz*qp2*qp2*S23
   -K*2d*qp1*qp2*S22
   -K3d*qp1*qp2*S2233
   +mb3z*2*qp1*qp2*l23z*S223
   -K3yz*qp3*qp3*S23
   -K3d*qp1*qp3*S2233
   +mb3z*2*qp1*qp3*l23z*C23*S2
   -K3yz*2*qp2*qp3*S23
```

6. Conclusions

A recursive method of determining the minimum set of dynamic parameters of a tree-like multibody system has been presented. Using simple rules, the number of independant dynamic parameters can be determined a priori. Depending on the mobility of the first joints of the kinematic chain, this number is equal to or less than $4n_{prismatic} + 7n_{revolute}$ while the total number of inertial parameters which is commonly used is given by $10(n_{prismatic} + n_{revolute})$. The reduction of independant parameters due to restricted mobility is also obtained from these rules in a straightforward manner. For instance, the first revolute joint of the Puma arm being vertical, it can be seen from table 4 that 9 parameters disappear from the dynamic model.

References

[1] H. Mayeda, K. Osuka & A. Kangawa, "A new identification method for serial manipulator arms" *IFAC 9th World Congress*, Budapest, July 2-6, 1984, pp 74-79

[2] M. Gautier, "Identification of robot dynamics", *Proc. IFAC Symposium on Theory of Robots*, Vienna, December 1986, pp. 351.

[3] C.G. Atkeson, C.H. An & J.M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links", *Int. J. of Robotics Research*, 5(3), 1986, pp 101-119

[4] E. P. Ferreira, "Contribution à l'identification des paramètres et à la commande dynamique adaptative des robots manipulateurs", *Thèse*, Université Paul Sabatier de Toulouse, 6 juillet 1984.

[5] P. Maes, J.C. Samin & P.Y. Willems, "Linearity of Multibody Systems with Respect to Barycentric Parameters : Dynamics and Identification Models Obtained by Symbolic Generation", *Mechanics of Structures and Machines*, Vol. 17, 1989, n°2, pp. 219-237

[6] J. Duelen & K. Schroër, "Robot calibration : Method and Results", *Robotic and Computer Integrated Manufacturing*, Vol. 8, pp. 223-231, 1991.

[7] P Maes, J.C. Samin & P.Y. Willems, "AUTODYN and ROBOTRAN , Computer Programmes", *Multibody Systems Handbook*, W. Schiehlen ed., 1990, pp. 225-264

[8] B. Raucent, J.C. Samin & P.Y. Willems, "Identification of Dynamic Parameters of Robot Manipulators from External Measurements", *European Journal of Mechanical Engineering*, Vol. 34, 1989, pp. 21-26.

[9] B. Raucent , G. Campion , G. Bastin, J.C. Samin & P.Y. Willems, "Identification of the Barycentric Parameters of Robot Manipulators from External Measurements", *Automatica*, 1992, vol. 28, n°5, pp. 1011-1016.

[10] B. Raucent & J.C. Samin, "Modeling and Identification of Dynamic Parameters of Robot Manipulators", *Robot Calibration*, R. Bernhardt and S. Albright ed., Chapman & Hall, to appear.

[11] E Dombre & W. Khalil, "Modélisation et commande des robots", Hermès, Paris, 1988

[12] J Wittenburg, *Dynamics of Systems of Rigid Bodies*, B. G. Teubner, Stuttgart, 1977

[13] M. Gautier, *"Contribution à la modélisation et à l'identification des robots"*, Thèse de doctorat d'état, Université de Nantes, 1990.

[14] Raucent B , *"Identification des parametres dynamiques des robots manipulateurs"*, Thèse de doctorat, Université Catholique de Louvain, 1990.

[15] Gautier M. & W. Khalil, "Direct Calculation of a Minimum Set of Inertial Parameters of Serial Robot", *IEEE Trans. on Robotics and Automation*, vol. 5-3, pp 368-372, 1990

[16] S. K Lin, "An identification method for estimating the inertial parameters of a manipulator", *Journal of Robotics Systems*, Vol. 9, 1992, n°4, pp 505-528.

[17] B Raucent & J.C Samin, 'Minimal Parametrization of Robot Dynamic Models', to appear in *Mechanics of Structures and Machines*

On the Use of Linear Graph Theory in Multibody System Dynamics

J.J. McPhee
Systems Design Engineering
University of Waterloo, Ontario
Canada N2L 3G1

ABSTRACT. Multibody dynamics involves the generation and solution of the equations of motion for a system of connected material bodies. The subject of this paper is the use of graph-theoretical methods to represent multibody system topologies and to formulate the desired set of motion equations; a discussion of the methods available for solving these differential-algebraic equations is beyond the scope of this work. After a brief introduction to the topic, a review of linear graphs and their associated topological arrays is presented, followed in turn by the use of these matrices in generating various graph-theoretic equations. The appearance of linear graph theory in a number of existing multibody formulations is then discussed, distinguishing between approaches that use absolute (Cartesian) coordinates and those that employ relative (joint) coordinates. These formulations are then contrasted with formal graph-theoretic approaches, in which both the kinematic and dynamic equations are automatically generated from a single linear graph representation of the system. The paper concludes with a summary of results and suggestions for further research on the graph-theoretical modelling of mechanical systems.

1. Introduction

A multibody system is hereby defined as *a finite number of material bodies connected in an arbitrary fashion by mechanical joints that limit the relative motion between pairs of bodies*. Practitioners of multibody dynamics study the generation and solution of the equations governing the motion of such systems. Several different approaches for systematically formulating these equations of motion have been developed, and encoded in numerical or symbolic computer programs. Such programs relieve an analyst of the error-prone labour involved in deriving the governing differential-algebraic equations by hand. For the program to be applicable to a wide range of mechanical systems, the topology of which is specified by the user at execution time and is not known *a priori*, the underlying formulation must somehow represent the connectivity of the system and use this topological information during the derivation of the equations of motion.

Linear graph theory is that branch of mathematics that studies the description and manipulation of system topologies. Interestingly, the extent to which this theory has been incorporated into current multibody dynamic formulations ranges from a bare minimum, typically in approaches that use absolute coordinates to describe the motion

of each body, to a maximum in formal graph-theoretic procedures It is the goal of this paper to examine this apparent conundrum in some detail, by explicitly referring to a number of existing formulations

The paper begins with a brief review of finite directed linear graphs and associated mathematical theorems, followed by a discussion of the minimal use of such graphs in absolute coordinate formulations. This in turn is followed by an examination of how linear graphs are used to represent system topologies in formulations that employ relative joint variables as generalized coordinates. Finally, a number of approaches that rely entirely on graph-theoretic techniques to automatically formulate the equations of motion are described. The paper concludes with a summary of results and suggestions for future exploitation of linear graph-theoretic methods in multibody dynamics.

2. Review of Graph-Theoretical Methods

The study of linear graph theory, one branch of the wider field of combinatorial mathematics, originated with Euler's famous solution [1] to the problem of the Königsberg bridges in 1736. A complete review of the contributions to linear graph theory since then is beyond the scope of this article. Instead, attention will be directed on those graph theorems that have application to the dynamic analysis of multibody systems. In consideration of the previous definition of a multibody system as a finite number of connected bodies, a discussion of infinite or unconnected graphs is excluded from this review. Furthermore, attention will be restricted to the study of directed graphs, since their undirected counterparts find less application in multibody dynamics. In summary then, the following review is limited to connected finite directed linear graphs and their associated mathematical theorems. For a more complete discussion of linear graph theory and its application to physical systems analysis, the reader is directed to references [2-5].

2.1 REPRESENTATION OF TOPOLOGY

A linear graph is a collection of edges, no two of which have a point in common that is not a vertex. In turn, an edge is defined as a line segment together with its distinct endpoints, and a vertex (or node) is an endpoint of an edge. The topology of a linear graph is completely defined when one specifies which edges are incident upon (connected to) each and every vertex of the graph.

To illustrate the use of linear graphs in representing topology, consider the diagram in Figure 1 which depicts an electrical network consisting of two inductors (L_1, L_2), three resistors (R_3, R_4, R_5), a capacitor (C_6), and a voltage source (V_7). Figure 2 shows the linear graph which is topologically equivalent, or isomorphic, to the electrical network. This graph is constructed by drawing a node for each point at which two physical elements connect, and by replacing these elements with directed edges on a one-to-one basis. The direction (arrow) assigned to each edge is arbitrary and serves only to provide a reference direction for "through" and "across" variables, which are defined in a subsequent section.

From the linear graph, one can construct the incidence matrix $[IN]$ which contains a complete topological description of the original physical system. This is a $v \times e$ matrix, where v is the number of vertices in the graph, and e is the number of edges

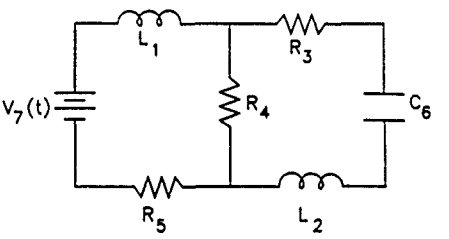


Figure 1: Electrical Network Example

The (i,j) entry in $[IN]$ is equal to $(-1,+1,0)$ if edge j is (negatively, positively, not) incident upon node i. The directed edge j is positively incident upon node i if it points towards the node, and negatively incident if it is directed away from the node. As an example, the incidence matrix for the linear graph in Figure 2 takes the form:

$$[IN] = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{ccccccc} L_1 & L_2 & R_3 & R_4 & R_5 & C_6 & V_7 \end{array} \\ \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix} \end{array} \tag{1}$$

in which the vertices associated with rows and the edges associated with columns have been explicitly labelled. One can see from this example that the rows of $[IN]$ are not linearly independent, since their sum is a row containing all zeroes. The $(v-1) \times e$ reduced incidence matrix $[A]$ is obtained by deleting any one row of $[IN]$; the remaining rows of $[A]$ are linearly independent, and the vertex corresponding to the deleted row is referred to as the datum node.

An important concept in linear graph theory is that of a circuit, which is a subset of a graph (or subgraph) such that, on every vertex, there are incident exactly two edges. In other words, a circuit is a closed loop of edges. With this concept, one can define a spanning tree, which is a subgraph containing all the vertices of the original graph, but no circuits. The $(v-1)$ edges in a tree of the graph are called branches, whereas the remaining edges in the complement of the tree, or cotree, are known as chords. One possible spanning tree for the graph in Figure 2 has been drawn with thick lines and consists of edges L_1, L_2, R_3, R_4, and R_5. The remaining edges C_6 and V_7 comprise the chords of the cotree. Having defined the concept of a tree, two new

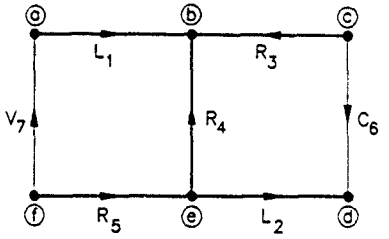


Figure 2: Linear Graph Isomorphic to Electrical Network

topological matrices can now be introduced — the fundamental cutset matrix and the fundamental circuit matrix.

A cutset is a subgraph of a connected linear graph such that, when deleted, the graph is left in two distinct parts. A fundamental cutset, or f-cutset, is a cutset consisting of one branch from the tree and a minimal number of chords such that no subgraph of the cutset is itself a cutset. As an example, the f-cutset corresponding to branch L_1 in Figure 2 consists of L_1 and V_7 (note that node a is not deleted along with the cutset edges and constitutes one of the two remaining parts of the graph). There will be $(v-1)$ f-cutsets for a given linear graph and tree, from which one can construct the **fundamental cutset matrix** $[CU]$. The (i,j) entry of $[CU]$ is zero if edge j is not in the f-cutset corresponding to branch i, and $(-1,+1)$ if edge j is in the f-cutset and has the (opposite, same) orientation as branch i. With this definition, the $(v-1) \times e$ matrix $[CU]$ takes the general partitioned form:

$$[CU] = \left[[1_t] \,\middle|\, [A_c] \right] \tag{2}$$

in which $[1_t]$ is a square unit matrix corresponding to the tree branches in the first $(v-1)$ columns, and $[A_c]$ is the remaining $(v-1) \times (e-v+1)$ submatrix. For the linear graph in Figure 2, the fundamental cutset matrix takes the specific form:

$$[CU] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{3}$$

An important result from linear graph theory is that the fundamental cutset matrix $[CU]$ is row-equivalent to the reduced incidence matrix $[A]$. In other words, $[CU]$

need not be constructed using its definition — it can be obtained *through simple row operations* on $[A]$. Verification of this theorem for the linear graph in Figure 2 is left to the reader.

A fundamental circuit, or f-circuit, is a circuit in which one edge is a chord from the cotree of the graph, and all the remaining edges are branches. To illustrate, the f-circuit corresponding to chord C_6 in Figure 2 also contains branches L_2, R_3, and R_4. There will be $(e - v + 1)$ f-circuits for a given graph and tree, from which the **fundamental circuit matrix** $[CI]$ can be constructed. The (i, j) entry of $[CI]$ is zero if edge j is not in the f-circuit corresponding to chord i, and $(-1, +1)$ if edge j is in the f-circuit and has the (opposite, same) direction as chord j as one travels around the circuit. With this definition, the $(e - v + 1) \times e$ matrix $[CI]$ can be written in the general form:

$$[CI] = \left[[B_t] \,\middle|\, [1_c] \right] \tag{4}$$

where $[1_c]$ is a square unit matrix corresponding to the cotree chords, and $[B_t]$ is the remaining $(e - v + 1) \times (v - 1)$ submatrix. The specific form of $[CI]$ for the example in Figure 2 is:

$$[CI] = \begin{bmatrix} 0 & -1 & -1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 1 \end{bmatrix} \tag{5}$$

Another important linear graph theorem [6] states that the rows of the circuit and cutset matrices for a given graph and tree are orthogonal, or mathematically,

$$[CI][CU]^T = [0] \tag{6}$$

in which the superscript T represents the transpose operation, and $[0]$ is a $(e - v + 1) \times (v - 1)$ matrix containing all zeroes. Substituting equations (2) and (4) into equation (6) and re-arranging, one can show that:

$$[B_t] = -[A_c]^T \tag{7}$$

which can be verified for the linear graph in Figure 2 by examining the specific form of equations (3) and (5).

In summary, the incidence matrix $[IN]$ contains a complete topological description of a given linear graph. The nonsingular reduced incidence matrix $[A]$ is obtained by deleting a row of $[IN]$ corresponding to the datum node. By selecting a spanning tree and numbering the first $(v - 1)$ columns of $[A]$ to correspond to the branches, the fundamental cutset matrix $[CU]$ can be obtained through simple row operations. This topological matrix explicitly shows $(v - 1)$ independent ways in which the given linear graph can be cut into two distinct parts — a result that is used later in this paper to generate the force equilibrium equations for mass elements in a multibody system without constructing the corresponding free-body diagrams. The fundamental circuit matrix $[CI]$ can, in turn, be obtained directly from $[CU]$ using equation (7) to construct the $[B_t]$ submatrix. Upon completion, $[CI]$ can be used to identify an independent set of $(e - v + 1)$ circuits in the given graph, an application of particular relevance to the kinematic analysis of multibody systems with closed loops.

2.2 GRAPH-THEORETICAL EQUATIONS

The circuit, cutset, and incidence matrices have more than topological significance — they can also be used to generate the governing equations for the physical system to which the linear graph is isomorphic. This fact was recognized almost four decades ago by Trent [5], who introduced the concept of through and across variables in order to set up these equations.

A **through variable** is a physical variable, associated with an edge of a graph, that would be measured by an instrument placed in series with the corresponding element in the original network. In the electrical network example in Figure 1, an appropriate through variable would be the current through an element. For a multibody system, the relevant through variables are forces or torques. Regardless of the type of physical system, the above definition of a through variable guarantees that the "Vertex Postulate", which states that *the sum of through variables at any node of a system graph must equal zero when due account is taken of the orientation of edges incident upon that node*, is satisfied for the linear graph. Essentially, the Vertex Postulate represents a generalization of Kirchoff's Current Law, from electrical network theory, that is applicable to all physical systems. It can be expressed mathematically for all v nodes in a graph by premultiplying the column matrix of through variables $\{\tau\}$ by the incidence matrix:

$$[IN]\{\tau\} = \{0\} \tag{8}$$

where $\{0\}$ is a $v \times 1$ column matrix of zeroes.

To obtain a linearly independent set of equations in the through variables, $[IN]$ in equation (8) can be replaced by either the reduced incidence matrix $[A]$ or the cutset matrix $[CU]$. In the latter case, and using equation (2), the resulting equations can be solved explicitly for the $(v-1)$ through variables $\{\tau_t\}$ associated with the branches:

$$\{\tau_t\} = -[A_c]\{\tau_c\} \tag{9}$$

in which $\{\tau_c\}$ are the cotree through variables. If these latter variables are known, then $\{\tau_t\}$ can be calculated directly from equation (9). It is for this reason that the cotree and tree through variables are referred to as "primary" and "secondary" variables, respectively. Using equation (3) for the electrical network example, equation (9) takes the specific form:

$$\begin{Bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \end{Bmatrix} = \begin{Bmatrix} -\tau_7 \\ \tau_6 \\ \tau_6 \\ -\tau_6 + \tau_7 \\ \tau_7 \end{Bmatrix} \tag{10}$$

which is equivalent to the set of equations obtained by successive applications of Kirchoff's Current Law to the network.

An **across variable** associated with an edge is a physical variable that would be measured by an instrument placed in parallel with the corresponding element in the physical network. In electrical systems, voltage is the most common example of such variables, whereas in multibody systems, relative displacements, velocities, and accelerations are all appropriate across variables. In general, the derivatives and integrals

of across variables are themselves across variables, as are vectors whose scalar components correspond to across variables. A rigorous definition of across variables is that they satisfy the "Circuit Postulate", namely that *the sum of across variables around any circuit of a graph must equal zero when due account is taken of the direction of edges in the circuit.*

To obtain a linearly independent set of equations that are mathematically equivalent to the Circuit Postulate, one need only premultiply a column matrix of across variables $\{\alpha\}$ by the circuit matrix, and set the result to $\{0\}$:

$$[CI]\{\alpha\} = \{0\} \tag{11}$$

Equation (4) can be used to express the $(e - v + 1)$ across variables associated with the chords $\{\alpha_c\}$ as explicit functions of the tree across variables $\{\alpha_t\}$:

$$\{\alpha_c\} = -[B_t]\{\alpha_t\} \tag{12}$$

which is why $\{\alpha_t\}$ is added to the list of primary variables, and $\{\alpha_c\}$ is a subset of the secondary variables. Using equation (5) for the electrical network example, equation (12) takes the form:

$$\begin{Bmatrix} \alpha_6 \\ \alpha_7 \end{Bmatrix} = \begin{Bmatrix} \alpha_2 + \alpha_3 - \alpha_4 \\ -\alpha_1 + \alpha_4 + \alpha_5 \end{Bmatrix} \tag{13}$$

which is equivalent to the independent set of equations obtained by applying Kirchoff's Voltage Law to the inner two loops of the network.

With e primary variables and another e secondary variables, the total number of unknowns is $2e$. Available for the solution of these unknown quantities are the $(v-1)$ cutset equations (9) and the $(e - v + 1)$ circuit equations (12) — a total of e linearly independent equations. The remaining e equations required to effect a solution are obtained from the constitutive, or terminal, equations for each of the e elements in the system. The terminal equation for each element expresses the functional relationship between the through and across variables and the independent variable, time. As an example, the terminal equation for the ideal inductor L_1 shown in Figure 1 would be:

$$\alpha_1 = L_1 \frac{d\tau_1}{dt} \tag{14}$$

where α_1 corresponds to the voltage across L_1, and τ_1 represents the current passing through the inductor. For a three-dimensional multibody system, the through and across variables are spatial vectors and the corresponding terminal equations are vectorial relationships. Regardless of the type of system under study however, the circuit, cutset, and terminal equations comprise a necessary and sufficient set for the solution of all unknown variables as functions of time [6]. The advantage of using these formal graph-theoretic techniques over traditional methods of formulating the governing equations is twofold:

1. the topological equations are clearly separated from the constitutive equations for the elements comprising the physical system under study, and

2. there is a number of standard graph-theoretic formulations available in the literature [2-5] for systematically establishing the governing equations — the particular choice of formulation method is made based upon the desired final form of these equations.

Finally, one may add that the systematic nature of formal graph-theoretic approaches make them quite attractive if a computer implementation is desired.

Before ending this review of linear graph theory, an auxiliary set of nodal variables are introduced. Essentially, a nodal variable corresponds to a physical variable that would be measured by an instrument placed between (across) the relevant node and a reference datum node. For an electrical network, the voltage difference between a node and the ground would be classified as a nodal variable. In a multibody system the absolute displacement, velocity, or acceleration of a point relative to an inertial reference frame all represent nodal variables. The reduced incidence matrix $[A]$ provides a convenient means for transforming between the nodal $\{\eta\}$ and across $\{\alpha\}$ variables of a system graph:

$$\{\alpha\} = [A]^T\{\eta\} \tag{15}$$

The nodal transformation equation (15) increases the total number of available equations by $(v-1)$, which corresponds to the additional $(v-1)$ unknowns, the auxiliary nodal variables.

3. Absolute (Cartesian) Coordinate Formulations

Attention is now switched to the other main topic of this paper, the dynamics of multibody systems. An example of such a system is the planar four-bar mechanism shown in Figure 3, which consists of three rigid links connected by ideal revolute joints to form a closed loop with the ground (the fourth "bar"). Assuming that a time-varying torque $T(t)$ is applied to link 1, the goal of a dynamic analysis would be to determine the configuration of the system at each instant during a finite interval of time. To represent the configuration, a set of coordinates that uniquely describe the position of each body in the system must be defined. The dynamic analysis is accomplished by generating the differential-algebraic equations (DAEs) of motion and solving them for the time-varying coordinates. This procedure has been successfully automated in a number of existing multibody programs using a variety of different formulation approaches.

Several of these approaches make use of **absolute coordinates**, which describe the position and orientation of each body in the system relative to an inertial frame of reference. For body 2 of the planar four-bar mechanism, shown in Figure 4, the absolute coordinates are defined as the Cartesian coordinates (x_2, y_2) of the mass center C_2 in the inertial XY reference frame, as well as the angle θ_2 between the link and the positive X axis. With three absolute variables per body, the total number of system coordinates $\{p\}$ is nine. Obviously, these nine coordinates are not independent since the four-bar mechanism has only one degree of freedom. The motion of individual bodies is constrained by the presence of kinematic joints between adjacent links; relationships between the absolute coordinates are obtained from a consideration of these holonomic constraint elements. Specifically, each pin joint requires that connected end

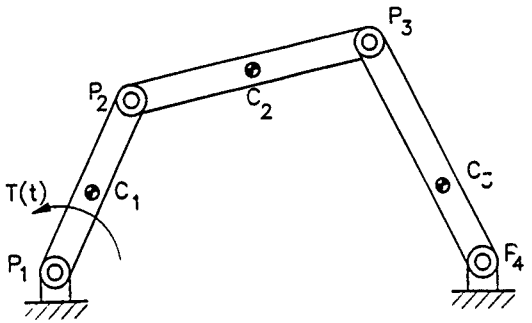


Figure 3: Torque-Driven Planar Four-Bar Mechanism

points of adjacent bodies remain coincident during any motion of the system. Mathematically, for the kinematic pair consisting of bodies $i-1$ and i connected by pin joint P_i:

$$\Phi_{2i-1} = x_{i-1} + \frac{l_{i-1}}{2}\cos\theta_{i-1} - x_i + \frac{l_i}{2}\cos\theta_i = 0 \tag{16}$$

$$\Phi_{2i} = y_{i-1} + \frac{l_{i-1}}{2}\sin\theta_{i-1} - y_i + \frac{l_i}{2}\sin\theta_i = 0 \tag{17}$$

noting that l_i is the length of uniform link i, the terms l_0, θ_0, l_4, and θ_4 are constants defining the positions of pins P_1 and P_4 in the inertial reference frame, and x_0, y_0, x_4, and y_4 are all identically zero.

Assembling equations (16) and (17) for the four pin joints results in a set of eight nonlinear algebraic equations:

$$\{\Phi(p)\} = \{0\} \tag{18}$$

which, by themselves, are not sufficient to solve for the nine absolute coordinates. To obtain a second set of equations for the system, one can write the equations of motion for each body in the system, by applying the Newton-Euler equations to a free-body diagram of each link. Note that the reaction forces $\{R\}$ in the pin joints enter explicitly into these expressions. Assembling the nine differential equations for all three bodies into a matrix form gives:

$$[M_a]\{\ddot{p}\} + [J]^T\{R\} = \{Q_a\} \tag{19}$$

in which $[M_a]$ is the generalized mass matrix, $\{Q_a\}$ contains the applied forces and torques, and $[J]$ is the Jacobian matrix of the constraint equations (18). In explicit

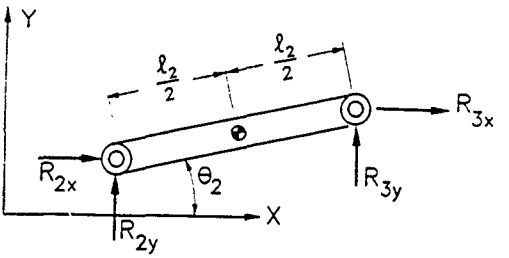


Figure 4: Free-Body Diagram of Link 2

form, the entries of $[J]$ are given by:

$$J_{ij} = \frac{\partial \Phi_i}{\partial p_j} \tag{20}$$

Together, equations (18) and (19) constitute a set of seventeen differential-algebraic equations that can be solved for the nine absolute coordinates $\{p\}$ and eight reaction forces $\{R\}$ as functions of the independent variable, time.

The approach described above has been extended and applied to the analysis of three-dimensional systems of rigid and flexible bodies connected by a wide variety of general holonomic and non-holonomic joints [7-13]. Due to its simplicity, this formulation can be readily implemented in a computer program that automatically generates the governing DAEs for a given mechanical system. A library of constraint equations for a variety of kinematic joints can be created in advance and used to generate the relationships between the absolute coordinates of two bodies connected by a particular type of joint. A second library containing the corresponding Jacobian matrix for each joint can also be created, and used to generate the system Jacobian matrix $[J]$ by means of an assembly process similar to that employed by the finite element method. The generalized mass matrix $[M_e]$ and forcing vector $\{Q_e\}$ can be constructed directly from the input data provided by a user. Thus, all terms in the governing equations (18) and (19) can be obtained *without using methods from linear graph theory.* As Haug [9] has observed, the use of absolute coordinates in the underlying formulation results in a "bypassing [of] topological analysis".

Orlandea et al [10] have used an even larger set of absolute coordinates than that described above in order to maximize the sparsity of the resulting DAEs, which are subsequently solved using sparse matrix methods and an implicit numerical integration routine. It is worth noting that these authors make explicit reference to the similarity between their derived equations and those arising in linear graph theory; specifically, they observe that equation (19) from their paper represents an assembly of the circuit, cutset, and terminal equations that would be obtained from a network model of their

multibody system. In spite of this comparison, the authors assemble their equations without exploiting the systematic nature of a graph-theoretical approach. Géradin [11] uses absolute coordinates in a finite element formulation of the equations of motion for flexible multibody systems. Once again, the use of such coordinates eliminates the need for linear graph theory since, as the author points out, "the topology of the articulated system is automatically embedded into its finite element description."

4. Relative (Joint) Coordinate Formulations

A second means of representing the time-varying configuration of a multibody system is through the use of **relative coordinates**. These coordinates describe the relative position and orientation of two adjacent bodies in terms of variables associated with the kinematic joint connecting the two bodies. A joint that allows j degrees of freedom (where $j \leq 6$) contributes j variables to the set of system relative coordinates $\{q\}$. As an example, the relative rotation between any two links in Figure 3 can be described by a single angle ϕ corresponding to the revolute joint connecting the two links. The complete set of three relative coordinates for this four-bar mechanism can be defined in terms of the previous absolute coordinates:

$$\begin{aligned} \phi_1 &= \theta_1 \\ \phi_2 &= \theta_2 - \theta_1 \\ \phi_3 &= \theta_3 - \theta_2 \end{aligned} \tag{21}$$

corresponding to pin joints P_1, P_2, and P_3. Note that it is not necessary to define the position of link 4 (the ground) relative to link 3.

If the multibody system has no closed loops, i.e. it has a "tree" topology, the set of relative coordinates is equal in number to the degrees of freedom of the system. As an example, if pin P_4 were removed from the four-bar mechanism, the resulting open-loop system would have three degrees of freedom. Using relative coordinates $\{q\}$, the equations of motion for such a system can be written in the form:

$$[M_r]\{\ddot{q}\} = \{Q_r(q, \dot{q}, t)\} \tag{22}$$

noting that reaction forces and torques in the joints have been eliminated from these equations using either an analytical approach (eg. d'Alembert's Principle or Lagrange's equations) or a systematic substitution process [14] in conjunction with the Newton-Euler equations. Equation (22) constitutes a set of ordinary differential equations that can be numerically integrated to obtain the relative coordinates as functions of time.

These same equations can be used to derive the DAEs governing the motion of a multibody system with closed loops. In this case, the relative coordinates are no longer independent and are related by the set of nonlinear algebraic constraint equations:

$$\{\Psi(q)\} = \{0\} \tag{23}$$

corresponding to the equations guaranteeing closure of an independent set of loops. For the four-bar mechanism, these loop closure equations can be obtained by summing the eight joint constraint equations (16) and (17), resulting in:

$$l_0 \cos\theta_0 + l_1 \cos\theta_1 + l_2 \cos\theta_2 + l_3 \cos\theta_3 - l_4 \cos\theta_4 = 0 \tag{24}$$

$$l_0 \sin\theta_0 + l_1 \sin\theta_1 + l_2 \sin\theta_2 + l_3 \sin\theta_3 - l_4 \sin\theta_4 = 0 \qquad (25)$$

Defining two new constants C_x and C_y that represent the distance between pins P_1 and P_4 in the X and Y directions respectively,

$$C_x = l_4 \cos\theta_4 - l_0 \cos\theta_0 \qquad (26)$$

$$C_y = l_4 \sin\theta_4 - l_0 \sin\theta_0 \qquad (27)$$

and using equations (21) to replace the absolute coordinates with their relative counterparts, the constraint equations (24) and (25) can be written in the desired form:

$$\Psi_1 = l_1 \cos\phi_1 + l_2 \cos(\phi_1 + \phi_2) + l_3 \cos(\phi_1 + \phi_2 + \phi_3) - C_x = 0 \qquad (28)$$

$$\Psi_2 = l_1 \sin\phi_1 + l_2 \sin(\phi_1 + \phi_2) + l_3 \sin(\phi_1 + \phi_2 + \phi_3) - C_y = 0 \qquad (29)$$

The final set of DAEs is obtained by removing from the system model one kinematic joint for each independent closed loop. The motion of the resulting open-loop system is governed by equation (22). By adding to this equation terms corresponding to the reaction forces and torques in the joints that were removed, the differential equations for the original system with closed loops are obtained:

$$[M_r]\{\ddot{q}\} + [K]^T\{\lambda\} = \{Q_r\} \qquad (30)$$

where $\{\lambda\}$ is a set of Lagrange multipliers, and $[K]$ is the Jacobian matrix of the loop closure constraint equations:

$$K_{ij} = \frac{\partial \Psi_i}{\partial q_j} \qquad (31)$$

Equations (23) and (30) constitute the final set of DAEs for the multibody system, expressed in terms of relative coordinates. One can see that they are similar in form to the corresponding equations (18) and (19) in absolute coordinates, but are generally fewer in number. The generalized mass matrix $[M_r]$ is smaller and less sparse than its absolute counterpart $[M_a]$, and the generalized forcing vector $\{Q_r\}$ is a complex function of the relative coordinates, their derivatives, and time. Even though the loop closure equations (23) have a higher order of nonlinearity than the constraint equations (18), the smaller set of DAEs for relative coordinates can be solved more efficiently than its counterpart in absolute coordinates. However, as Nikravesh [8] has pointed out, this computational advantage is offset by the additional labour required to generate these equations.

One source of this additional labour is the requirement for topological processing when relative coordinates are employed. The reason for this is simple: regardless of the type of approach one uses to derive the differential equations (30), one needs expressions for the velocity (eg. to formulate the kinetic energy for use with Lagrange's equations) or the acceleration (eg. using the Newton-Euler equations) of each body *relative to an inertial frame of reference*. For a particular body m_i, one can obtain these expressions from the relative coordinates and their time derivatives only if one knows the identity and ordering of the intermediate bodies connecting m_i to the inertial reference frame. In brief, a mathematical description of the system topology is required, such a description is conveniently provided by linear graph theory

that takes the general form:

$$[W] = \left[\begin{array}{c|c} [S_0] & [S_0^*] \\ \hline [S] & [S^*] \end{array}\right] \tag{32}$$

or specifically, for the linear graph of the four-bar mechanism:

$$[W] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ \hline -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & -1 \end{array}\right] \tag{33}$$

Upon examination, one can see that the matrix $[W]$ is exactly the negative of the incidence matrix $[IN]$ defined in Section 2 of this paper. The first row of $[W]$ corresponds to the datum node, and the $(*)$ superscript is used to identify hinges in the cotree of the graph. Therefore, the $n \times n$ submatrix $[S]$, where n is one less than the number of nodes (bodies) in the system, represents the tree portion of the reduced incidence matrix $[A]$. The inverse of this matrix was first introduced by Branin [17] and called the "node-to-datum-path matrix", since it can be constructed by examining which edges iie in the path between a given node and the datum node.

Unaware of Branin's previous work, Roberson and Wittenburg [18] re-invented this matrix, called it simply the "path matrix" $[T]$, and verified the relationships:

$$[T]^T[S_0]^T = -\{1\} \tag{34}$$

$$[T][S] = [S][T] = [1] \tag{35}$$

where $\{1\}$ is a column matrix of n elements, each equal to 1, and $[1]$ is the $n \times n$ identity matrix. Wittenburg [14] makes extensive use of the $[S]$ and $[T]$ matrices to derive the differential equations (30) for a multibody system. Unfortunately, the generation of the constraint equations (23) is only presented by way of three example analyses. This drawback was subsequently addressed in [19], in which Wittenburg presents a systematic procedure for formulating the loop closure equations, once again making use of the topological matrices $[S]$ and $[T]$.

It is interesting to note that formal graph-theoretic methods can be used to derive many of the equations appearing in Wittenburg's formulation. To do so, one must identify a set of through and across variables [5] that respectively satisfy the vertex and circuit postulates for Wittenburg's linear graph. The relative rotational velocities $\{\Omega\}$ form a natural set of across variables for this graph, and must therefore satisfy the fundamental circuit equation (12):

$$\{\Omega_c\} = -[B_t]\{\Omega_t\} \tag{36}$$

where $\{\Omega_c\}$ and $\{\Omega_t\}$ correspond to hinges in the cotree and tree, respectively. Recalling equation (7) and using the interesting fact [17,19] that the path, reduced incidence, and cutset matrices are related by:

$$[T][S^*] = [A_c] \tag{37}$$

Sheth and Uicker [15] recognized this fact over two decades ago, when they used Hamilton's Principle and relative coordinates to formulate the differential equations for a multibody system consisting entirely of closed kinematic loops. To obtain the supplementary set of loop closure constraint equations, the authors construct a simple linear graph representation of the multibody system, in which nodes represent links and edges represent kinematic joints. With this linear graph, the authors are able to derive an independent set of loop closure equations that are equivalent to the fundamental circuit equations, with the relative coordinates playing the role of across variables. Furthermore, a graph-theoretical algorithm is employed to minimize the number of joints appearing in the final set of independent loops. It is possible to identify further equivalences between the equations derived by Sheth and Uicker and the graph-theoretical circuit and cutset equations; the interested reader is referred to the thesis by Li [16]

In the first monograph on multibody dynamics, Wittenburg [14] makes direct use of linear graph theory to represent the topology of a system of rigid bodies containing both open and closed kinematic loops. Similar to the model employed by Sheth and Uicker, rigid bodies are represented as nodes of the graph and hinges between bodies appear as edges. Wittenburg's graph model is more encompassing however, since his "hinge" has been generalized to include springs, dampers, and other connections having six degrees of freedom. In addition, Wittenburg makes explicit use of a "spanning tree" that is consistent with conventional graph-theoretic methods. For the four-bar mechanism, the linear graph thus defined takes the form shown in Figure 5. The four nodes have been labelled as m_i, where m_i represents link i, and the hinges labelled P_j correspond to the four pin joints. The edges comprising an arbitrarily-selected tree are drawn as solid lines, while the one cotree edge is dashed.

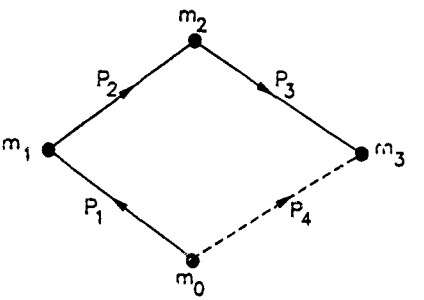


Figure 5: Wittenburg's Linear Graph of Four-Bar Mechanism

To represent the system connectivity, Wittenburg defines an incidence matrix [W]

one can verify that equation (36) is identical to Wittenburg's equation (5.211), which is used to find an expression for the virtual work done in cut (cotree) hinges.

An appropriate nodal variable for Wittenburg's graph is the difference between the absolute angular velocity of each body and that of the base body (node):

$$\eta_i = \underline{\omega}_i - \underline{\omega}_0 \tag{38}$$

A subset of the nodal transformation equation (15) can be used to relate this set of nodal variables $\{\eta\}$ to the tree across variables $\{\Omega_t\}$, if one recognizes that $-[S]$ corresponds to the tree portion of the reduced incidence matrix $[A]$:

$$\{\Omega_t\} = -[S]^T(\{\underline{\omega}\} - \underline{\omega}_0\{1\}) \tag{39}$$

Premultiplying equation (39) by $-[T]^T$, one obtains Wittenburg's equation (5.122):

$$\{\underline{\omega}\} = -[T]^T\{\Omega_t\} + \underline{\omega}_0\{1\} \tag{40}$$

which is subsequently used to obtain the differential equations (22) for the open-loop (tree) portion of the multibody system. For a further discussion of the relationships between Wittenburg's formulation and formal graph-theoretic methods, the reader is directed to the forthcoming paper [20].

If one is only interested in using the topological description to generate kinematic relationships, then alternatives to a linear graph representation are available in the literature. As an example, the tree topology of an open-loop multibody system is represented by Huston and Passerello [21] using a "body connection array", which is subsequently employed by Amirouche [22] to generate the loop closure equations for a general system of rigid and flexible bodies. Nikravesh and Gim [23] transform the absolute coordinate DAEs (18) and (19) to the set of equations (23) and (30) in relative coordinates using a topology-dependent velocity transformation matrix; Kim and Vanderploeg [24] have presented a systematic procedure for constructing this matrix using a modified version of the path matrix $[T]$, in which all -1's are replaced by $+1$'s. Pereira and Proença [25] have applied a similar transformation to the DAEs for a multibody system containing rigid and flexible members. Hiller et al [26] have presented an alternative means of transforming the equations from absolute to relative to (in some instances) minimal coordinates, using the concept of a "kinematical transformer" to represent a closed loop; the topology of the complete system is contained in a block diagram of these kinematical transformer elements.

Using a linear graph representation similar to that employed by Wittenburg, Hwang and Haug [27] have developed a recursive formulation for a system of rigid bodies with closed loops, and implemented this formulation on a computer with parallel processing capabilities. In a subsequent formulation for flexible multibody systems, Lai et al [28] achieved a greater degree of parallelism by defining an "extended graph" in which nodes represent reference frames attached to bodies and edges represent the transformations between these frames. It is interesting to note that this graph representation is very similar to the "vector-network model" first introduced by Andrews and Kesavan [29], which is discussed further in the next section.

5. Graph-Theoretic Multibody Formulations

In the previous section, it was shown how a number of current multibody dynamics formulations use linear graph theory to represent the system topology and to generate various kinematic relationships. The difference between such approaches and a formal graph-theoretic procedure has been summarized by Li [16]:

> "To appreciate the difference between an ad hoc application of graph theory to dynamics and a graph-theoretic approach, one must understand that a terminal graph (edge) is part of a mathematical model of a physical component. If the individual system components are modelled properly, the resulting system graph will automatically satisfy the cutset and circuit postulates."

As discussed in Section 2, if each edge is associated with a physical component and a consistent set of through and across variables is chosen, then a graph-theoretic formulation will automatically provide a necessary and sufficient set of DAEs. In the formulations discussed in the previous section, the edges did not correspond to physical components but simply represented the connectivity of the system. It is impossible to obtain the dynamic equations of motion directly from such a linear graph.

One of the first applications of formal graph-theoretic methods to multibody dynamics resulted in the "vector-network method" of Andrews and Kesavan [29]. This graph-theoretic formulation was used to systematically derive the equations of motion for a three-dimensional system of particle masses. More recently, it has been extended to the analysis of two-dimensional [16] and three-dimensional [30] constrained mechanical systems. To better understand this formulation, consider the vector-network model of the four-bar mechanism shown in Figure 6. In this graph representation, edges represent the position vectors corresponding to physical components while nodes represent connection points between these components. In addition to representing the three rigid bodies therefore, the edges m_1, m_2, and m_3 locate the mass centers of these links relative to a datum node ⓓ located in inertial space. Similarly, edges r_5 and r_6 represent the points where pins P_1 and P_4 attach to the ground. The "rigid-arm" elements s_4 to s_{11} correspond to body-fixed position vectors from the mass centers to the points on each body where the revolute joints are connected. The driving torque is represented by the edge T_{12}, while the remaining edges h_{13} to h_{16} correspond to pin joints P_1 to P_4, respectively.

The vector-network graph is significantly more complex than its counterpart in Figure 5 because it contains dynamic, as well as kinematic, information. The existence of torque and mass elements in the vector-network model is evidence of this. Of course, the vector-network graph also contains the topological information provided by Wittenburg's graph, which can be recovered from Figure 6 by deleting the dynamic mass and torque elements, and contracting each combination of kinematic joint and connected rigid-arm elements into a single edge.

To generate the system equations using a graph-theoretic approach, a tree is selected that contains elements 1 to 11, with the torque and kinematic joint elements placed in the cotree. A suitable across variable is the position vector $\underline{r}_i$ corresponding to element i in the graph, while the force $\underline{F}_i$ in this element is identified as an appropriate through

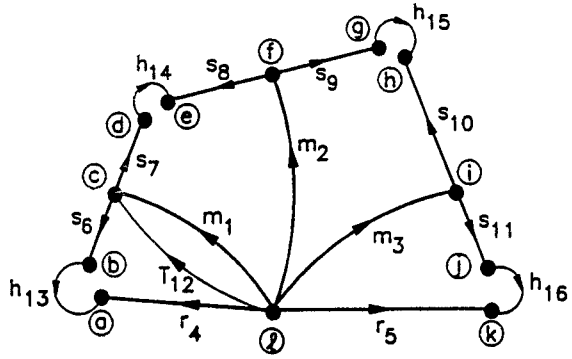


Figure 6: Vector-Network Graph of Four-Bar Mechanism

variable. For each pin joint element h_i, one can write the terminal equation:

$$\mathfrak{x}_i \equiv 0 \tag{41}$$

recognizing the fact that the two points (nodes) connected by a revolute joint remain coincident for any possible motion of the system. With this terminal equation, the kinematic constraint equations can be systematically generated using the fundamental circuit equations (12) for the cotree pin joints. As an example, the circuit equation corresponding to kinematic joint h_{14} (P_2) is:

$$\mathfrak{x}_{14} = \mathfrak{x}_8 + \mathfrak{x}_2 - \mathfrak{x}_1 - \mathfrak{x}_7 = 0 \tag{42}$$

The eight constraint equations (16) and (17) are obtained by resolving the vectorial circuit equations for the four pin joints into components parallel to the inertial X and Y axes. In a similar systematic fashion, and recognizing the fact that the terminal equations for the mass elements correspond to the vectorial Newton-Euler equations, the cutset equations for these tree elements result in nine scalar differential equations in which the reaction forces in the pin joints appear explicitly. One can see that, as a consequence of using absolute coordinates to represent the position and orientation of each mass element, the final set of DAEs generated by the vector-network method corresponds exactly to equations (18) and (19).

Using an analytical substitution procedure, Li [16] has reduced this set of DAEs to the smaller set corresponding to equations (23) and (30). More recently, Bacıu and Kesavan [31] have derived these latter equations directly using a linear graph model similar to that employed by Wittenburg, relative coordinates, and a formal

graph-theoretic approach. As discussed in Section 2 and demonstrated above, such an approach provides a systematic formulation procedure in which the topological equations are clearly distinguished from the constitutive equations. This has the important implication that new types of mechanical elements, modelled by suitable terminal equations, can be incorporated into the multibody system model without altering the underlying formulation procedure. With this in mind, efforts are currently underway to incorporate nonholonomic joints, flexible bodies, and variable-mass elements into existing vector-network formulations. One further advantage of a graph-theoretic approach is that it can provide a single unified formulation for systems containing elements from different physical regimes, eg. hydraulic, pneumatic, or electronic elements. This leads the author to be optimistic about the future application of the vector-network method to the growing field of mechatronics.

6. Conclusions

To develop a dynamic formulation that is applicable to general multibody systems with open and closed loops, some means of representing the topology of the system is required. If one uses absolute coordinates to describe the location of bodies in the system, the topological description is automatically embedded in the relatively large set of DAEs. If one uses a set of relative (joint) coordinates however, some form of topological processing is required in order to account for the connectivity of the bodies. To represent the system topology and to generate some of the required kinematic relationships, several researchers have made use of linear graphs and their associated topological matrices. It has been shown that many of these kinematic relationships are equivalent to the equations obtained from a formal graph-theoretic approach. The advantages of using such an approach have been discussed, and the systematic generation of the governing DAEs using the vector-network method has been described. Research is underway to extend this graph-theoretical method to the analysis of flexible multibody and mechatronic systems.

7. Acknowledgements

Financial support by an NSERC Research Grant is gratefully acknowledged, as are the efforts of Lorraine Kritzer in speedily and meticulously preparing the figures in this paper.

8. References

[1] N.L. Biggs, E.K. Lloyd, and R.J. Wilson, Graph Theory: 1736-1936, Oxford University Press, 1976

[2] S. Seshu and M.B. Reed, Linear Graphs and Electrical Networks, Addison-Wesley, 1961.

[3] R.G. Busacker and T.L. Saaty, Finite Graphs and Networks: An Introduction with Applications, McGraw-Hill, 1965.

[4] H.E. Koenig, Y. Tokad, and H.K. Kesavan, **Analysis of Discrete Physical Systems**, McGraw-Hill, 1967.

[5] H.M. Trent, *Isomorphisms between Oriented Linear Graphs and Lumped Physical Systems*, J. Acous. Soc. Am., vol.27 (1955), pp.500-527.

[6] G.C. Andrews, *A General Restatement of the Laws of Dynamics Based on Graph Theory*, in **Problem Analysis in Science and Engineering**, edited by F.H. Branin, Jr. and K. Huseyin, Academic Press, 1977, pp.1-40.

[7] P.E. Nikravesh and E.J. Haug, *Generalized Coordinate Partitioning for Analysis of Mechanical Systems with Nonholonomic Constraints*, ASME J. Mech. Trans. Auto. Design, vol.105 (1983), pp.379-384.

[8] P.E. Nikravesh, **Computer-Aided Analysis of Mechanical Systems**, Prentice-Hall, 1988.

[9] E.J. Haug, **Computer-Aided Kinematics and Dynamics of Mechanical Systems, Volume 1**, Allyn and Bacon, 1989.

[10] N. Orlandea, M.A. Chace, and D.A. Calahan, *A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems — Parts 1 and 2*, ASME J. Eng. Industry, vol.99 (1977), pp.773-784.

[11] M. Géradin, *Computational Aspects of the Finite Element Approach to Flexible Multibody Systems*, in **Advanced Multibody System Dynamics**, edited by W. Schiehlen, Kluwer Academic Publishers, 1993, pp.337-354.

[12] A.A. Shabana, **Dynamics of Multibody Systems**, Wiley and Sons, 1989.

[13] A. Avello and J. García de Jalón, *Dynamics of Flexible Multibody Systems Using Cartesian Co-ordinates and Large Displacement Theory*, Int. J. Num. Meth. Eng., vol.32 (1991), pp.1543-1563.

[14] J. Wittenburg, *Dynamics of Systems of Rigid Bodies*, B.G. Teubner Stuttgart, 1977.

[15] P.N. Sheth and J.J. Uicker, Jr., *IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage*, ASME J. Eng. Industry, vol.94 (1972), pp.454-464.

[16] T.W. Li, **Dynamics of Rigid Body Systems: A Vector-Network Approach**, M.A.Sc. Thesis, University of Waterloo, Canada, 1985.

[17] F.H. Branin, Jr., *The Relation between Kron's Method and the Classical Methods of Network Analysis*, Matrix and Tensor Quarterly, vol.12 (1962), pp.69-105.

[18] R.E. Roberson and J. Wittenburg, *A Dynamical Formalism for an Arbitrary Number of Interconnected Rigid Bodies, with Reference to the Problem of Satellite Attitude Control*, Proceedings of 3rd IFAC Congress, Vol.1, Book 3, Paper 46D, Butterworth, 1966.

[19] J. Wittenburg, *Graph-Theoretical Methods in Multibody Dynamics*, Contemporary Mathematics, vol. 97 (1989), pp.459-468.

[20] M. Ishac, J. McPhee, and G.G. Andrews, *Wittenburg's Formulation of Multibody Dynamics Equations from A Graph-Theoretic Perspective*, submitted to First International Conference on Graphs and Mechanics, Poland, October, 1993.

[21] R.L. Huston and C. Passerello, *On Multi-Rigid-Body System Dynamics*, Comp. Structures, vol. 10 (1979), pp.439-446.

[22] F.M.L. Amirouche, Computational Methods in Multibody Dynamics, Prentice-Hall, 1992.

[23] P.E. Nikravesh and G. Gim, *Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops*, Proceedings of ASME Design Automation Conference, 1989, pp.27-33.

[24] S.S. Kim and M.J. Vanderploeg, *A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations*, ASME J. Mech. Trans. Auto. Design, vol.108 (1986), pp.176-182.

[25] M.S. Pereira and P.L Proença, *Dynamic Analysis of Spatial Flexible Multibody Systems Using Joint Co-ordinates*, Int. J. Num. Meth. Eng., vol.32 (1991), pp.1799-1812.

[26] M. Hiller, A. Kecskemethy, and C. Woernle, *A Loop-Based Kinematical Analysis of Complex Mechanisms*, ASME Paper 86-DET-184, 1986.

[27] R.S. Hwang and E.J. Haug, *Topological Analysis of Multibody Systems for Recursive Dynamics Formulations*, Mech. Structures and Mach., vol.17 (1989), pp.239-258.

[28] H.J. Lai, E.J. Haug, S.S. Kim, and D.S. Bae, *A Decoupled Flexible-Relative Coordinate Recursive Approach for Flexible Multibody Dynamics*, Int. J. Num. Meth. Eng., vol.32 (1991), pp.1669-1689.

[29] G.C. Andrews and H.K. Kesavan, *The Vector-Network Model: A New Approach to Vector Dynamics*, Mech. Mach. Theory, vol.10 (1975), pp.57-75.

[30] G.C. Andrews, M.J. Richard, and R.J. Anderson, *A General Vector-Network Formulation for Dynamic Systems with Kinematic Constraints*, Mech. Mach. Theory, vol.23 (1988), pp 243-256.

[31] G. Bacu and H.K. Kesavan, *From Unidimensional to Multidimensional Physical Systems: Graph-theoretic Modeling*, Systems and Cybernetics, vol 21 (1992), pp.55-71.

A RELATIONAL DATABASE FOR GENERAL MECHANICAL SYSTEMS

C. HARDELL, A. STENSSON AND P. JEPPSSON
Division of Computer Aided Design
Department of Mechanical Engineering
Luleå University of Technology
S - 971 87 Luleå
SWEDEN

ABSTRACT. This paper provides a specification of a relational database structure for mechanical systems. Through the example provided, a robot gripping device, it is demonstrated how the initial development of the database structure has successfully led to an implementation in a practical software environment. The database is accessible to multiple engineering application programs and supports a flexible environment for the continuing development of new applications.

1. Introduction

The engineering process when designing and manufacturing mechanical systems is now going through substantial changes. The changes are intended to shorten development time, to improve quality and to reduce cost of products. Specialized disciplines are integrated in a simultaneous engineering process where all the relevant activities such as design, analysis, testing and manufacturing are involved in the early stages of the development of the product. The benefit of this technique is the ability to make correct decisions in the early stages of the design of a product. This of course reduces re-work, improves quality and saves money compared with the traditional way of design where for example simulations of product performance were carried out in a late stage of the product development time when changes are expensive.

To ease engineering, several computer aids have been developed. Important tools are programs for Computer Aided Design (CAD), Finite Element Analysis (FEA), Computer Aided Manufacturing (CAM), Multibody System Analysis (MSA) and special purpose simulations. These tools can be used successfully one by one but the large benefits are gained first after an integration where data created or changed in one application immediately can be used in the other steps of the development process. For example it should be possible to use the geometry of a solid model, already defined in a CAD program, when creating a FE model or generating NC-code in a CAM program. The administration of data is preferably carried out with the product data stored in a central database which is possible to access from all applications, see Figure 1.

In this paper modern database techniques are demonstrated in the analysis and design of mechanisms. In particular the generality and the easy access and alteration of the data in the database system are considered. The paper focuses on the product database and especially the parts which are important for multibody system analysis. See Figure 2 where "Simulation of Performance" from Figure 1 has been divided into "Strength of Material" and "Multibody System Analysis".

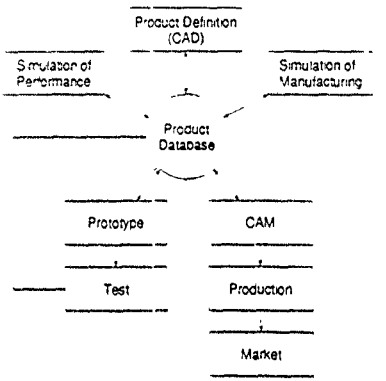


Figure 1 Information flow of a computer aided simultaneous engineering system

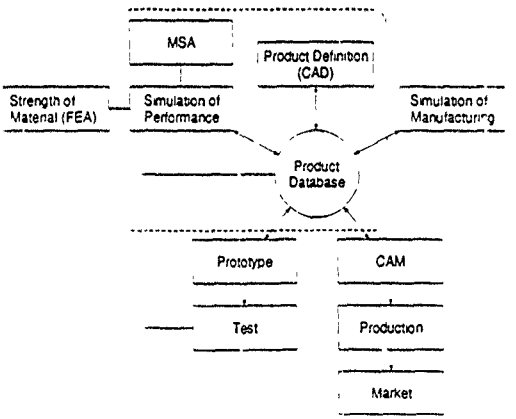


Figure 2 The main area of interest of this paper

2. Databases for Mechanical Systems

When designing mechanical systems as for example vehicles, robots and NC-machines an important aspect of the function of the product is its dynamical behaviour. It is therefore important that simulation and analysis software for the dynamical behaviour of a product is integrated in a design system. A huge number of methods and software have been developed which allow the automatic formulation and solving of the equations of motion by the computer for many types of mechanical systems. Summaries of computer codes for simulating multibody dynamics are given in [1] and [2]. The multibody systems are built of rigid and/or flexible bodies with inertia and springs, dampers and force actuators, interconnected by joints, rigid bearings and supports Friction forces, contact forces and constraints may also be included. Information about these components can be included in the database.

If mechanical systems are to be stored in a general form in a database the appropriate and sufficient information about the mechanism has to be defined. The data of the mechanical system is preferably defined independently of the formalism applied for the generation of equations of motion and are extracted from the geometrical description of the system. Since newer and better tools are developed rapidly it is important that the design engineers are able to change computer tools whenever necessary. To make the design system robust and independent of the specific application programs, the product database should be independent of the applications which are accessing it.

Using a common database for all application programs (codes) also reduces the number of required interfaces between programs. The effort when adding a new application is then substantially reduced since the application only needs to communicate with the database.

Using the geometry data and the resulting time histories of the dynamical behaviour of the mechanical system, the motion can be animated. This is a favourable approach for debugging of the mathematical model, to obtain an overall idea of the motion and for pinpointing specific problems such as improper connectivity, misapplied motions or forces, unwanted oscillatory behaviour and clearance problems. Therefore the geometry (curves, surfaces etc.) should also be easily accessible from the database.

2.1 MECHANISM DATABASE STRUCTURE AND TABLES - AN EXAMPLE

The demands above can lead to the database structure of Figure 3. In this database structure, a mechanism is defined as a collection of rigid bodies that can move relative to each other, with joints that limit relative motion of pairs of bodies. A reference triad is a local coordinate system that consists of an origin and three axes, x, y and z. Each reference triad is attached to a rigid body at a selected location. The location and orientation of a joint are defined by a pair of reference triads associated with two rigid bodies. The location of a force is defined by a reference triad Stiffnesses and dampings between rigid bodies are defined between pairs of reference triads.

The motion of a mechanism is driven by loads. The loading may be gravity, applied forces, applied motion and initial conditions. The variations of applied forces and applied motions can be described by input functions. The geometry of each rigid body is defined in the object table and its subtables. Observe that this database structure is flexible and can be changed according to the user's desire.

Tables corresponding to the database structure in Figure 3 are shown in Figure 4. At the top of the tables the table name are located, followed by the column names. The coupling between the tables are made by key records which are marked with lines. The different tables contain records of the different datatypes, integer, character, real and double precision.

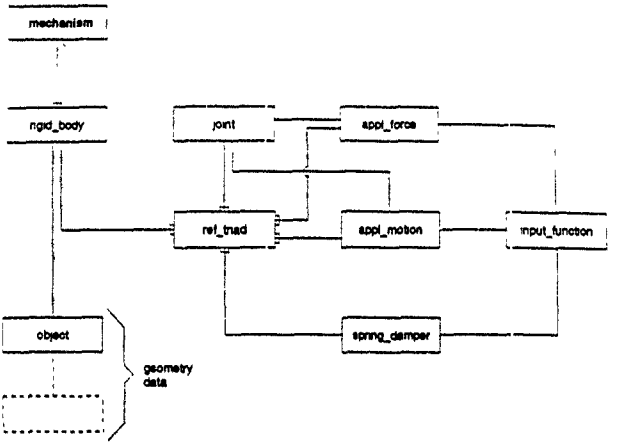


Figure 3. The mechanism database structure. A single line illustrates a one to one relationship, for example the geometry of one rigid body is described by one object definition. A fork illustrates a one to several relationships, for example a mechanism consists of more than one rigid body.

As an example the table *joint* is described below. The descriptive information in the table *joint* consists of six records. *joint_no* is an integer value identifying the specific joint. *joint_nam* is a character string containing the name of the specific joint. *ref_triad_1* and *ref_triad_2* are integer values pointing at the two reference triads in the *ref_triad* table associated with the joint. *joint_type* is an integer value describing which type of joint it is, examples are revolute joints (1), slider joints (2), ball joints (3), cylindrical joints (4) and planar joints (5). The last record *jt_user_flag*, is an integer flag that is reserved for user specific information.

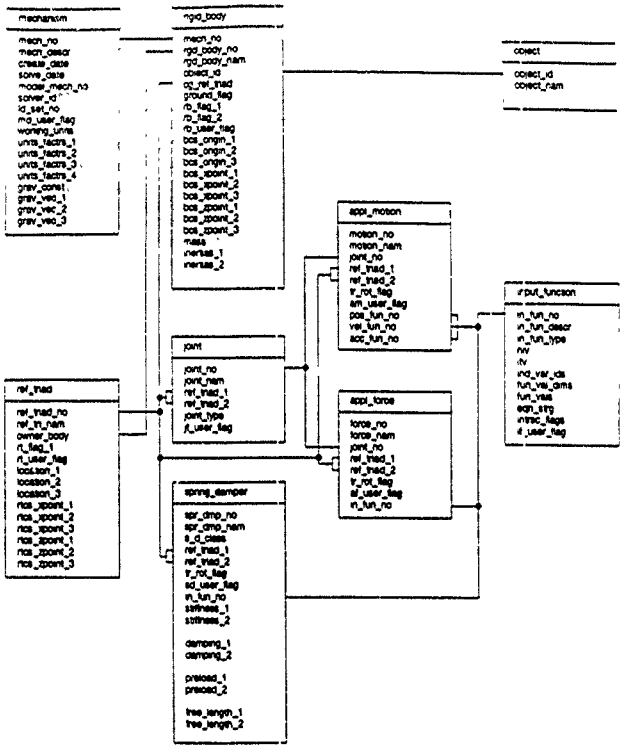


Figure 4 Tables defining a mechanical system

2.2 ESSENTIAL CHARACTERISTICS OF ENGINEERING DATABASE MANAGEMENT SYSTEMS

When engineering database management systems are to be chosen, some essential characteristics have to be considered [3-7].

An engineering database must be accessible to multiple engineering application programs and it must also support an environment for the continuing development of new applications. Application programs use different representations of the same objects to achieve the most efficient solution to the particular problem they are designed to solve. This implies that the database schemes must be flexible enough to support multiple views of the data in the database as well as modification and extension of the schemes. The manipulation of data must include the operations insert, delete and query. Data models for design of database schemes should preferably support engineering data types (vectors, matrices etc.) and the complex object types found in engineering applications.

An engineering database system should provide a facility for managing integrity constraints on the database and should automatically maintain a consistent database state. The control of consistency of the data should be included in the applications or in the executive of the system. The executive is a command processor that assembles information, launches application programs, accumulates performance predictions, and provides control over the process by the user [3-5].

In order to support access of the database by engineering application codes, the database system must have a host language interface. Because of the amount of engineering computing that is done in Fortran, the interface should include access for at least this language. The system executive should use query languages embedded in the host language to access the information in the database. Standard query languages facilitate the extraction of information even for a none expert. All types of data (administrative, mechanical etc.) should be available with the same query language.

A product development system requires a system executive that controls the execution of application programs a database management system, user interfaces, support utilities, networking capabilities and distributed computing. The executive system must support multiple and concurrent users, who may operate either in batch mode or interactively.

The database system should not impose constraints on a preferred sequence of activities in the design process. Concurrent users must have access to the system simultaneously independently of their activities. This is one of the foundations of simultaneous engineering [8].

2.3 THE CHOSEN SYSTEM SOLUTION

Our solution to the demands above is to develop the database using the relational database management system called INFORMIX [9,10]. The flexibility of the relational data model is the important property making it adaptable to the needs of engineering database management [4]. INFORMIX consists of useful programs or modules that perform data management tasks. The manipulation of the data is performed using the standard high level languages SQL and 4GL [10]. INFORMIX supports distributed processing which allows users across the network to easily access the same database.

The database structure described in Figure 3 and 4 has been implemented in the chosen relational database management system as part of a software development project, carried out at the Division of Computer Aided Design at Luleå University of Technology, Sweden.

The handling of consistency maintenance can be divided into two parts. First there is consistency check when inserting data into the database. Records that must have a value are not allowed to be null. For example a joint must have a joint number, two reference triads and a joint type. If a non-consistency should occur it is eliminated through an administrative function that deletes all information with non-valid relations.

The communication between the database, the applications and the user interface are taken care of and controlled by a central executive.

3. Example - a Robot Gripping Device

The following example illustrates how the proposed system operates when processing a typical mechanical system. Consider the development of a gripping device for a robot. The function of the device is to grip and lift a cylinder with a diameter of 50 mm and a mass of 5 kg, see Figure 6. The first stage is to identify the geometry of the individual parts in the mechanism. This can be done in any suitable CAE system We have chosen to define the gripping device using the program I-DEAS by SDRC [11].

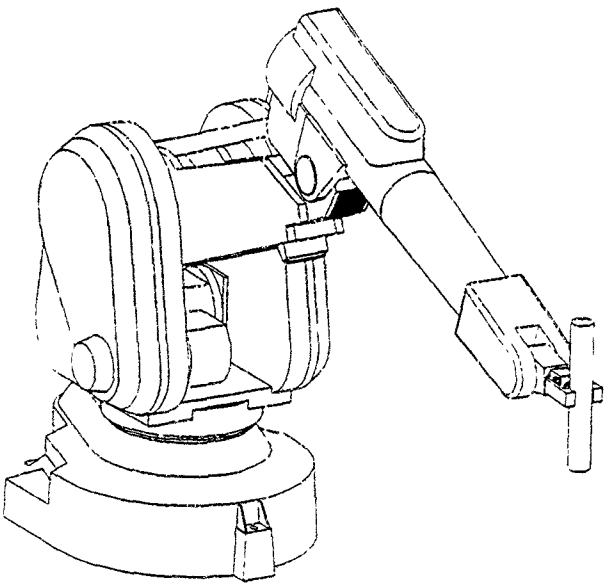


Figure 6 A robot with a gripping device

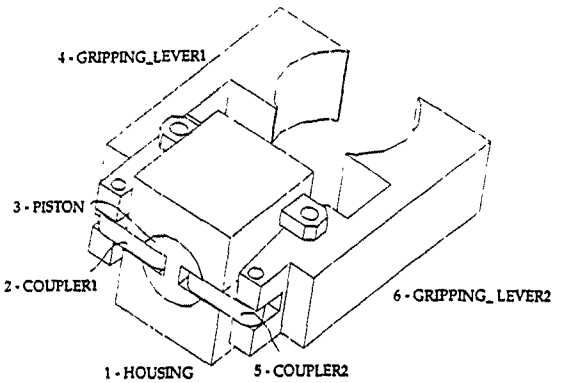
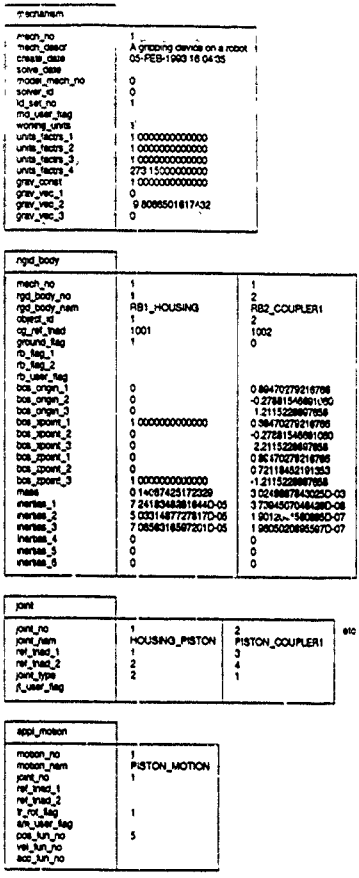


Figure 7 The robot gripping device.

The gripping device shown in Figure 7 consists of six objects. The objects are modeled and their corresponding physical characteristics are defined. Then the device is assembled and its joints, applied motion and applied forces are added to the system. The bodies are connected by seven joints. Joint number one is a slider joint between PISTON and HOUSING and the six other joints are revolute joints connecting GRIPPING_LEVER1 with COUPLER1, GRIPPING_LEVER1 with HOUSING, COUPLER1 with PISTON, COUPLER2 with PISTON, GRIPPING_LEVER2 with HOUSING and finally COUPLER2 with GRIPPING_LEVER2. Here, the piston is forced by an applied motion which is proportional to time. When the mechanism is defined the analysis can be made using different computer codes, depending on the specific tasks. In this example the task may be to make sure that no parts are interfering, that the main function is fulfilled i.e., gripping a 50 mm cylinder and that the design will have enough strength.

To be able to make the analysis in any suitable software the necessary information of the mechanism is transferred to the neutral format described in Figure 3 and Figure 4. This is done by reading from the relational database PEARL within the I-DEAS system to the mechanism database developed in INFORMIX as described above. Examples on data in the tables in the product database for this mechanism are shown in Figure 8. The application programs may extract the suitable information from the database using the query language. The interfaces can now be written in embedded Fortran or 4GL [10] for automatic transfer of the data to a suitable format for the application program.



mechanism

mech_no	1
mech_descr	A gripping device on a robot
create_date	05-FEB-1993 16:04:35
solve_date	
master_mech_no	0
solver_id	0
id_set_no	1
md_user_flag	
working_units	1
units_factrs_1	1.0000000000000
units_factrs_2	1.0000000000000
units_factrs_3	1.0000000000000
units_factrs_4	273.15000000000
grav_const	1.0000000000000
grav_vec_1	0
grav_vec_2	-9.8066501617432
grav_vec_3	0

rigid_body

mech_no	1	1	etc
rgd_body_no	1	2	
rgd_body_nam	RB1_HOUSING	RB2_COUPLER1	
obeid_id	1	2	
cg_ref_frad	1001	1002	
ground_flag	1	0	
rb_flag_1			
rb_flag_2			
rb_user_flag			
bas_origin_1	0	0.89470279216766	
bas_origin_2	0	-0.27881546691950	
bas_origin_3	0	1.21115228867658	
bas_zpoint_1	1.0000000000000	0.38470279216766	
bas_zpoint_2	0	-0.27881546691950	
bas_zpoint_3	0	2.21115228867658	
bas_zpoint_1	0	0.85470279216766	
bas_zpoint_2	0	0.72118452191353	
bas_zpoint_3	1.0000000000000	-1.21115228867658	
mass	0.14097425172329	3.02488878430230-03	
inertias_1	7.24183483818440-05	3.77845079484980-08	
inertias_2	5.03314877278170-05	1.90124--1508850-07	
inertias_3	7.08563165972010-05	1.98050208955670-07	
inertias_4	0	0	
inertias_5	0	0	
inertias_6	0	0	

joint

joint_no	1	2	etc
joint_nam	HOUSING_PISTON	PISTON_COUPLER1	
ref_frad_1	1	3	
ref_frad_2	2	4	
joint_type	2	1	
rj_user_flag			

appl_motion

motion_no	1
motion_nam	PISTON_MOTION
joint_no	1
ref_frad_1	
ref_frad_2	
r_rot_flag	1
a/s_user_flag	
pos_fun_no	5
vel_fun_no	
acc_fun_no	

Figure 8 Examples on the tables in the product database.

4 Discussion and Conclusions

Simultaneous engineering demands a full integration of engineering computer tools. The integration is preferably carried out with the product data stored in a central database, a system executive that controls the execution of application programs, a database management system, user interfaces, support utilities, networking capabilities and distributed computing.

The product data of a mechanical system is preferably defined independently of the formalism applied for the generation of equations of motion and are extracted from the geometrical description of the system. Since newer and better tools are developed rapidly it is important that the design engineers are able to change computer tools when necessary. To make the design system robust and independent of the specific programs, the product database should be independent of the applications accessing it. Using a common database for all application programs also reduces the number of required interfaces between programs. The effort when adding a new application is then substantially reduced since the new application only needs to communicate with the database.

This paper provides a specification of a relational database structure for mechanical systems. Through the example provided, it demonstrates how the initial development of the database structure has successfully led to an implementation in a practical software environment. The database is accessible to multiple engineering application programs and support a flexible environment for the continuing development of new applications.

5. Acknowledgments

The authors wish to thank the Swedish Board for Technical and Industrial Development and the CIM Institute at Luleå University of Technology for the financial support.

6. References

1 W. Kortüm, R. S. Sharp and A. D de Pater. *Application of Multibody Computer Codes to Vehicle System Dynamics* Progress Report to the 12th IAVSD Symposium on a Workshop and Resulting Activities, Society for Engineering and Scientific Education, 1991

2 W Schiehlen. *Multi Body Systems Handbook*. Springer-Verlag 1990

3 S A. Barley and R. J. Cripps. *Executive-Centred System Design for CAD Applications* Computer-Aided Design 24 (1992) 5 pp 235-242.

4 S M. Staley and D C. Anderson. *Functional Specification for CAD Databases*. Computer Aided Design 18 (1986) pp 132-138

5 B. Dopker. *Developments in Interdisciplinary Simulation and Design Software for Mechanical Systems*. Engineering with Computers 4 (1988) pp 229-238

6 W Beitz and J. Feldhusen. *Management Systems and Program Concepts for an Integrated CAD Process*. Research in Engineering Design 3 (1991) pp 61-74

7 A J Midland, G. Mullineux, I. R. B Potts, R. Singh and E. Sittas. *Interfacing Personal Designer with External Software* Proc. Computervision European Users' Conference, Brussels, 1990

8 L. Karlsson, H-Å. Häggblad, C. Hardell, P. Jeppeson and B-O. Elfström. *Simultaneous Engineering can be Realized by the Aids of Integrated Design and Manufacturing* Teaching note, Division of Computer Aided Design, Luleå University of Technology, Sweden.

9 INFORMIX. Informix Software Inc 4100 Bohannon Drive, Menlo Park, CA 94025 USA

10 J Leffler. *Using INFORMIX-SQL* Addison Wesley 1991

11 I-DEAS SDRC, Structural Dynamics Research Corporation, 200 Eastman Drive, Milford, Ohio 45150-2789 USA

A METHOD FOR LINEARIZATION OF THE DYNAMIC EQUATIONS OF FLEXIBLE MULTIBODY SYSTEMS

Y. LI and C. GONTIER
LMSS, Ecole Centrale des Arts et Manufactures
Grande Voie de Vignes
92295 Chatenay-Malabry Cedex
FRANCE

ABSTRACT We propose a simple method for the linearization of the nonlinear dynamic equations, which are established in our case based on the Kane's virtual work principle. The system generalized variables used in the equations are the reference coordinates at the connections for the rigid body motions and the time derivatives of the body modal coordinates for the flexible body behaviors. In the linearization process, the partial derivations of the kinematic terms with respect to the generalized coordinates and to the generalized velocities have to be performed. In our method, these partial derivatives are yielded through the first and second order time derivations of the body rotational absolute angular velocities and through the first, second, and third order time derivations of the mass center vectors, respectively. This is the key point of our method. The partial derivation of the mechanical terms, e.g., of the inertial tensors etc., are also calculated. We have developped the specific operators to perform the time derivations without distinguishing the rigid and flexible variables. This method makes both the theoretical deduction and the programming relatively simple, and allows a rapid computing.

1. Introduction

Many mechanical and structural systems such as vehicules, manipulators and aircrafts, consist of interconnected components that undergo large rotations. The characteristic of such flexible multibody systems is the high degree of nonlinearity of its dynamics. An accurate mathematical model for these systems has to account for inertial, centrifugal, and Coriolis nonlinearities due to large body motion. The tendency, especially in aircraft and robotic industries, to produce systems with lightweight components that operate at very high speeds enhances the demand for such accurate mathematical models. These systems incorporate various types of driving, sensing, and controlling devices working together to achieve specified performance requirements. However, many of these systems exhibit serious vibrations that cause damage to the system components, at least a deviation from a required performance. The dynamic analysis and control design of these nonliner systems are thus important, which is however complicated due to the coupling between the nonlinear rigid body motions and the linear elastic displacements of the body during the motion. In general, the behaviours of such systems can be accurately reproduced by the nonlinear dynamic equations. It is also possible to use the nonlinear dynamic model as the core of a control algorithm but it is very expensive in computing [1, 2]. Thus a linear dynamic model for the nonlinear system is usually required in order to simplify the computing and to adapt to the control design which has been developped for the linear

systems and already exists as well as modal analysis. Obviously to reach this purpose, the linearization procedure is necessary to be performed

In this paper, we propose a simple linearization method to obtain the linear dynamic model for a flexible multibody nonlinear system Our nonlinear dynamic equations for the system are established using the Kane's virtual work principle. Elastic bodies in the system are discretized using the finite element approach. Their deformation modes are directly derived from the differential equations with appropriate boundary conditions. A finite element code can be used for this calculation (e.g.,The Nastran or Systus software).The configuration of each elastic body is defined by using a set of absolute coordinates and modal coordinates. The kinematic relations between the interconnected bodies are represented by the recusive equations [3]. Obviously, a direct partial derivation of these nonlinear dynamic equations in the linearization procedure can generate a large amount of terms, due to the strong coupling and nonlinear relationships among the variables, which can hardly be handled in the programming, especially on performing the partial derivations of kinematic terms with respect to the generalized coordinates in order to establish the stiffness matrix of the linearized system. The derivation process is thus overwhelming.

However, we can in fact obtain these partial derivatives through the first and second order time derivations of the body absolute angular velocities and through the first, second, and third order time derivations of the mass center vectors, respectively, so that the linearization procedure can be very much simplified. We have developed the specific operators to perform the time derivatives without distinguishing the rigid and flexible variables. We like to describe our nonlinear elastic model in Section 2, to introduce our linearization procedure in Section 3, to present our sequence of computation in Section 4 and to give a numerical example in Section 5. We conclude in Section 6.

2. System Dynamic Equations of Motion

The Kane's virtual work principle [4, 5] states that the sum of the virtual works of elastic forces, of active forces and of inertia forces is equal to zero, all of which are acting on the system. Note that the non-working interconnected constraint forces between the bodies of the system are eliminated in using this principle. The principle can be expressed as:

$$\int_V \sigma \delta\varepsilon^T dV - \int_\Sigma \mathbf{F}\,\delta\mathbf{r} d\Sigma + \int_M \ddot{\mathbf{r}}\,\delta\mathbf{r} dM = 0. \qquad (1)$$

V is the volume of the body k in the deformed state. $\delta\mathbf{r}$ is the kinematically admissible virtuel displacement of a point in the body, $\mathbf{r}$ the acceleration of the point, $\mathbf{F}$ the external force and the body force, and σ and ε represent the stress and strain tensors. In correspondence to Eqn (1), the principle can also be written as.

$$\delta W^{int} - \delta W^{ext} + \delta W^{ine} = 0 \qquad (2)$$

Among the three terms, the virtual work of the inertial force is the most delicate and complicate to develop. In the following, we give in some details the development of this term and only a brief mention of the two other terms.

2.1 GENERALIZED INERTIAL FORCES

We first develope the virtual work of inertial forces.

$$\delta W^{ine} = \int_M \ddot{r}\,\delta r dM$$

Figure 1 shows the undeformed (shadowed) and deformed states of a flexible body k. r and a are the position vector of an arbitary point M on the deformable body and the position vector of the mass center. X is the vectors from the mass center (G_d) to an arbitrary point M on the deformed body and Y is the vector from the mass center (G_o) to an arbitrary point M_o on the undeformed body. The following formulation uses an inertial reference frame (R_o) to describe the translation motion and a body-fixed frame (R_d) for the rotational motion.

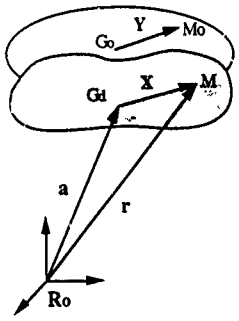


Fig. 1. Deformed (shadowed) and undeformed states of the flexible body.

The position of the point M on the deformable body can be written as:

$$r = a + X \tag{3}$$

$$X = Y + [U(M) - U(G)]$$

$$U(M) = \sum_{\alpha=1}^{n} \xi_\alpha \Phi_\alpha(M) \quad \text{and} \quad U(G) = \sum_{\alpha=1}^{n} \xi_\alpha \Phi_\alpha(G).$$

n is the number of modes for the deformed body Φ_α is the vector of the vibration shape and ξ_α is the modal coordinates.

Deriving Eqn (3) with respect to time yields

$$\mathbf{r} = \mathbf{a} + \boldsymbol{\omega} \Lambda \mathbf{X} + \overset{\circ}{\mathbf{X}}$$

$$\overset{\circ}{\mathbf{X}} = \hat{\mathbf{U}}(M) - \hat{\mathbf{U}}(G), \tag{4}$$

in which ω is the absolute angular velocity vector of the body k and $\overset{\circ}{\mathbf{X}}$ the time derivative in the body reference R_d.

The acceleration of the point M can be determined by the direct derivation of Eqn.(4), which is

$$\mathbf{r} = \mathbf{a} + \boldsymbol{\omega} \Lambda \mathbf{X} + \boldsymbol{\omega} \Lambda (\boldsymbol{\omega} \Lambda \mathbf{X}) + 2\boldsymbol{\omega} \Lambda \overset{\circ}{\mathbf{X}} + \overset{\circ\circ}{\mathbf{X}}$$

$$\overset{\circ\circ}{\mathbf{X}} = \hat{\mathbf{U}}(M) - \hat{\mathbf{U}}(G) \tag{5}$$

The virtual deplacements of the point M is

$$\delta \mathbf{r} = \delta \mathbf{a} + \delta \mathbf{X} = \delta \mathbf{a} + \delta \boldsymbol{\theta} \Lambda \mathbf{X} + \delta \xi_\alpha (\Phi_\alpha(M) - \Phi_\alpha(G)) \tag{6}$$

in which $\delta\theta$ is the virtual angular deplacement of the body k. Thus

$$\delta \mathbf{a} = \mathbf{V}_l \delta q_l + \mathbf{V}_\alpha \delta \xi_\alpha = \mathbf{V} \{\delta p\}^T \qquad (\alpha = 1, 2, \ldots n_f) \tag{7}$$

$$\delta \boldsymbol{\theta} = \boldsymbol{\Omega}_l \delta q_l + \boldsymbol{\Omega}_\alpha \delta \xi_\alpha = \boldsymbol{\Omega} \{\delta p\}^T \qquad (l = 1, 2, \ldots n_r) \tag{8}$$

$\mathbf{V} = [\mathbf{V}_l \ . \ \mathbf{V}_\alpha]$, $\boldsymbol{\Omega} = [\boldsymbol{\Omega}_l \ . \ \boldsymbol{\Omega}_\alpha]$, and $\{\delta p\}^T = \{\delta q_l, \delta \xi_\alpha\}^T$

n_r is the number of the rigid degree of freedom (DOF). n_f is the number of the flexible DOF. q_l is the relative joint coordinates for the rigid body motion. The vectors V and Ω are the partial velocity and the partial angular velocity of the body k which are all the complicated implicit functions of p. Their recusive expressions are given in detail in [3]

We thus have obtained the generalized inertial forces for the body k associated with δq_l and $\delta \xi_\alpha$:

$$\mathbf{F}_l^* = - \mathbf{V}_l \mathbf{M} \ \ddot{\mathbf{a}} - \boldsymbol{\Omega}_l \{\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \Lambda \mathbf{J}\boldsymbol{\omega} + \mathbf{K}\boldsymbol{\omega} + \ddot{\xi}_\alpha \mathbf{H}_\alpha\} \tag{9}$$

$$\mathbf{F}_\alpha^* = - \mathbf{V}_\alpha \mathbf{M} \ \mathbf{a} - \boldsymbol{\Omega}_\alpha \{\mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \Lambda \mathbf{J}\boldsymbol{\omega} + \mathbf{K}\boldsymbol{\omega} + \ddot{\xi}_\alpha \mathbf{H}_\alpha\}$$

$$- \{\mathbf{H}_\alpha \dot{\boldsymbol{\omega}} - \boldsymbol{\omega} \cdot \mathbf{J}_\alpha \boldsymbol{\omega} + \dot{\mathbf{H}}_\alpha \boldsymbol{\omega} + \mu_\alpha \ddot{\xi}_\alpha - \sum_{\beta=1}^{n} M \Phi_\alpha(G) \Phi_\beta(G) \ddot{\xi}_\beta\}, \tag{10}$$

in which

$$\omega = \Omega \dot{p}. \tag{11}$$

$$\dot{a} = v \dot{p}. \tag{12}$$

$$\dot{\omega} = \dot{\Omega} \dot{p} + \Omega \ddot{p}. \tag{13}$$

$$\dot{a} = \dot{v} \dot{p} + v \ddot{p}. \tag{14}$$

and M is the mass of the body k. We call μ_α, J, K, J_α, H_α, and $\mathring{H}_\alpha$ are the mechanical tensor terms of the deformed body whose definitions can be found in Appendix.

2.2 GENERALIZED ELASTIC AND EXTERNAL FORCES

The virtual work due to the elastic forces can be written as.

$$\delta W^{int} = - \int \sigma \delta \varepsilon^T dV = - \xi_\alpha^T K_{\alpha\alpha} \delta \xi_\alpha.$$

$K_{\alpha\alpha}$ is the symmetrical positive definite suffness matrix associated with the elastic model coordinates of the body k in the multibody system. The matrix $K_{\alpha\alpha}$ is very easy to be obtained by the finite -element method.

The virtual work of all the external forces acting on the body k can be also written as:

$$\delta W^{ext} = \int_F \delta r d\Sigma = \{Q\}^T \{\delta p\},$$

where $\{Q\}^T$ is the vector of the generalized force in function of the generalized coordinates

2 3 GENERAL FORM OF THE DYNAMIC EQUATION

We have developed the generalized forces. The dynamic equation for the flexible multibody system can thus be obtained by summing up the contributions of the generalized forces to Eqn.(1) over all the bodies. The general form of the equation is written as:

$$M(p) \ddot{p} + B(\dot{p}, p) + g(p) + d \dot{p} + kp = T, \tag{15}$$

in which. M(p) is the system mass matrix,

$B(\dot{p}, p)$ is the vector of the Coriolis and centrifugal forces,

g(p) is the vector of the gravitational force,
d is the system damping matrix,
k is the system stiffness matrix,
and T is the vector of the torques acting on the joints

It is evident that the dynamic equation for the flexible multibody system is nonlinear due to the strong coupling between the nonlinear rigid body motions and the linear elastic displacements of the body In order to modal analyze and to control design such a system, we need to perform the linearization of Eqn.(15).

3. Linearization of the Dynamic Equation

Linearizing Eqn.(15) about a point ($\dot{p}_o$, p_o, T_o) yields a linear equation as written below

$$M(p_o)\,\ddot{p} + D(\dot{p}_o, p_o)\,\dot{p} + K(\dot{p}_o, p_o)p = \bar{T} \tag{16}$$

in which D (= d + $\partial B/\partial \dot{p}$) is the damping matrix and K (= k + $\partial B/\partial p$ + $\partial g/\partial p$) the stiffness matrix for the linearized system. The linearization of Eqn.(15) consists therefore in computing the matrices M, K, and D.

It is necessary to perform the partial derivation for each term in B in Eqn (15) with respect to $\dot{p}$ and p, to establish the damping and stiffness matrices for the linearized system. Note that the generalized inertial force terms which contain p and p, as shown in Eqn (9) and (10), consist in the major part of the B vector. We rewrite these terms below:

$$b_l = V_l M\ \ddot{a}^\rho + \Omega_l \{J\dot{\omega}^\rho + \omega \Lambda J\omega + K\omega\} \tag{17}$$

$$b_\alpha = V_\alpha M\ a^\rho + \Omega_\alpha J\dot{\omega}^\rho + (\Omega_\alpha \omega\ J\omega) + \Omega_\alpha K\omega + H_\alpha \dot{\omega}^\rho \cdot \omega \cdot J_\alpha \omega + \dot{H}_\alpha \omega \tag{18}$$

in which $a^\rho = \dot{V}\dot{p}$ and $\dot{\omega}^\rho = \dot{\Omega}\dot{p}$.

In order to obtain ($\partial b_l/\partial p_j$), ($\partial b_\alpha/\partial \dot{p}_j$), ($\partial b_l/\partial p_j$), and ($\partial b_\alpha/\partial p_j$), we need both the p_j and p_j derivatives of the kinematic quantities such as a^ρ, $\dot{\omega}^\rho$, ω, V, Ω and the p_j derivatives of the mechanical quantities such as J, K, J_α, H_α, and $\dot{H}_\alpha$ to obtain the $\dot{p}_j$ and p_j derivatives of b_l and b_α. However the direct partial derivations of a^ρ, $\dot{\omega}^\rho$, ω, V, and Ω with respect to p and p are very long so that a large amount of terms is yielded which can hardly be handled in the programing. The complication mainly results from the derivations of the kinematic quantities. Thus, we introduce a simple method to fullfill these derivations

Let us write down the first, second, and third order time derivations for the mass center vectors of the body k They are all the implicit functions of p while the explicit functions of p, as written below

$$a_i = A_i(p)\,\dot{p}_i \qquad (i = 1, 2, \ldots N) \tag{19}$$

$$a_k = B_{ij}(p)\dot{p}_i p_j + C_i(p)\dot{p}_i \qquad (j = 1, 2, \ldots N) \tag{20}$$

$$a_k = D_{ijk}(p)\dot{p}_i p_j p_k + E_{ij}(p)\dot{p}_i \dot{p}_j + F_i(p)\dot{p}_i \qquad (k = 1, 2, \ldots N) \tag{21}$$

in which $A_i(p) = C_i(p) = F_i(p)$. $N\ (=n+n_r)$ is the total number of the generalized coordinates of the system. A_i, B_{ij}, C_i, D_{ijk}, E_{ij}, and $\vec{F}_i$ are the coefficient vectors, among which $B_{ij}(p)$ and $D_{ijk}(p)$ resulted from the partial derivations of the mass center vectors with respect to p_i, p_j, and p_k. According to the alternate property of the partial differentiation, we have:

$$\begin{aligned} &B_{ij}(p) = B_{ji}(p) \\ &D_{ijk}(p) = D_{jki}(p) = D_{kij}(p) = D_{ikj}(p) = D_{jik}(p) = D_{kji}(p). \end{aligned} \tag{22}$$

Similarly the first and second time derivatives for the absolute angular velocities are,

$$\omega_k = \vec{A}_i(p)\,\dot{p}_i \tag{23}$$

$$\dot{\omega}_k = \vec{B}_{ij}(p)\dot{p}_i p_j + \vec{C}_i(p)\dot{p}_i \tag{24}$$

$$\dot{\omega}_k = \vec{D}_{ijk}(p)\dot{p}_i p_j p_k + \vec{E}(p)_{ij}\dot{p}_i \dot{p}_j + \vec{F}_i(p)\ddot{p}_i, \tag{25}$$

in which, $\vec{A}_i(p) = \vec{C}_i(p) = \vec{F}_i(p)$. The $\vec{B}_{ij}(p)$ and $\vec{D}_{ijk}(p)$ vectors have the similar relations (Eqn.(22)). Comparing Eqn.(19), (20), (23), and (24) with Eqn.(11), (12), (13), and (14), respectively, we obtain:

$V_i = A_i(p)$,

$V_i = B_{ji}(p)\dot{p}_j$,

$\Omega_i = \dot{A}_i(p)$, and

$\Omega_i = \vec{B}_{ij}(p)\dot{p}_j$

Our idea of simplification of the linearization procedure is to make use of these coefficient vectors. We memorize these vectors, among which the $B_{ij}(p)$ and $D_{ijk}(p)$ vectors which satisfy the relations given by Eqn (22) are sumed up and put into the $B_{ij}(p)$ and $D_{ijk}(p)$ vectors for which i>j>k. The $\dot{p}$ and p derivatives of the kinematic terms can be formed by these coefficient vectors times the corresponding combination of the generalized velocities. The following formulae are established in such a way:

$$\frac{\partial v_i}{\partial p_k} = B_{ik}(p) \tag{26}$$

$$\frac{\partial a^p}{\partial p_k} = \sum_{i=1}^{k-1} B_{ik}(p)\dot{p}_i + 2B_{kk}(p)\dot{p}_k + \sum_{i=k+1}^{N} B_{ik}(p)\dot{p}_i \tag{27}$$

$$\frac{\partial \dot{a}^p}{\partial p_k} = \frac{\partial \dot{a}^p}{\partial p_k} = D_{kkk}(p)\dot{p}_k\dot{p}_k + \frac{2}{3}\{\sum_{i=1}^{k-1} D_{kki}(p) + \sum_{i=k+1}^{N} D_{iki}(p)\}\dot{p}_i\dot{p}_k$$

$$+ \frac{1}{3}\{\sum_{i=1}^{k-1}\sum_{j=1}^{i} D_{kij}(p) + \sum_{i=k+1}^{N}\sum_{j=1}^{k-1} D_{ikj}(p) + \sum_{j=k+1}^{N}\sum_{i=1}^{N} D_{ijk}(p)\}\dot{p}_i\dot{p}_j \tag{28}$$

Note that the p derivative of a^p is obtained by the p derivative of $\dot{a}^p$, which makes the derivation easier because $\dot{a}$ is the explicit function of $\dot{p}$ as mentioned earlier. The fractions in the third equation are resulted from such an indirect derivation.

$$\frac{\partial \Omega_i}{\partial p_k} = \bar{B}_{ik}(p) \tag{29}$$

$$\frac{\partial \omega^p}{\partial p_k} = \sum_{i=1}^{k-1} \bar{B}_{ik}(p)\dot{p}_i + 2\bar{B}_{kk}(p)\dot{p}_k + \sum_{i=k+1}^{N} \bar{B}_{ik}(p)\dot{p}_i \tag{30}$$

$$\frac{\partial \omega}{\partial p_k} = \sum_{i=1}^{k-1} \bar{B}_{ik}(p)\dot{p}_i + 2\bar{B}_{kk}(p)\dot{p}_k + \sum_{i=k+1}^{N} \bar{B}_{ik}(p)\dot{p}_i \tag{31}$$

$$\frac{\partial \omega}{\partial p_k} = \bar{A}_k(p) \tag{32}$$

$$\frac{\partial \omega^p}{\partial p_k} = \frac{\partial \dot{\omega}^p}{\partial p_k} = \bar{D}_{kkk}(p)\dot{p}_k\dot{p}_k + \frac{2}{3}\{\sum_{i=1}^{k-1} \bar{D}_{kki}(p)) + \sum_{i=k+1}^{N} \bar{D}_{iki}(p)\}\dot{p}_i\dot{p}_k$$

$$+ \frac{1}{3}\{\sum_{i=1}^{k-1}\sum_{j=1}^{i} \bar{D}_{kij}(p) + \sum_{j=1}^{k-1}\sum_{i=k+1}^{N} \bar{D}_{ikj}(p) + \sum_{j=k+1}^{N}\sum_{i=1}^{N} \bar{D}_{ijk}(p)\}\dot{p}_i\dot{p}_j \tag{33}$$

To sum up, all the $\dot{p}$ and p derivatives of the kinematic terms can be obtained by simply performing the time derivations for the mass center vectors and the absolute angular velocities instead of by performing a complicated direct derivation. In order to calculate

quickly the time derivation, we try to formulate the recurrent relationships among the bodies regarding to the mass center vectors and the absolute angular velocities

3.1 RECURRENCE EXPRESSION OF THE ANGULAR VELOCITIES AND THEIR DERIVATIONS WITH RESPECT TO TIME

In our recurrence expressions, we do not specify the generalized coordinates of the rigid and flexible bodies by using different notations. We simply use p_i ($i=1, 2, \ldots, N$) to coordinate the bodies. Thus we labled the system in the way as described in Fig. 2

The connecting points in the system are also labled as the bodies. The vector θ_{2k-1} at the connecting points represents the rotational direction of the rigid body in the system. The vector θ_{2k} on each body represents the rotational direction of the deformed body. Thus the absolute angular velocity of any body in the system can be expressed as:

$$\omega_k = \omega_{k-1} + p_{(k-1),\alpha}\,\theta_{(k-1),\alpha} \tag{34}$$

Note that α now represents the number of the rigid or flexible DOF. Derivating Eqn. (34), we can obtain:

$$\omega_k = \omega_{k-1} + \omega_{k-1} \Lambda (p_{(k-1),\alpha}\,\theta_{(k-1),\alpha}) + \dot{p}_{(k-1),\alpha}\,\theta_{(k-1),\alpha} \tag{35}$$

$$\omega_k = \omega_{k-1} + \omega_{k-1} \Lambda (p_{(k-1),\alpha}\theta_{(k-1),\alpha}) + \omega_{k-1} \Lambda (\omega_{k-1} \Lambda (p_{(k-1),\alpha}\theta_{(k-1),\alpha}))$$

$$+ 2\omega_{k-1} \Lambda (\dot{p}_{(k-1),\alpha}\theta_{(k-1),\alpha}) + \dot{p}_{(k-1),\alpha}\,\theta_{(k-1),\alpha} \tag{36}$$

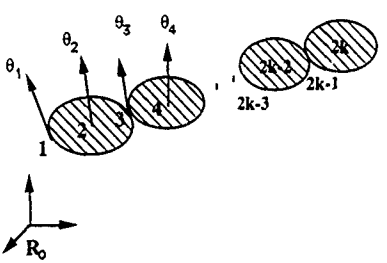


Fig 2 A rotational chain

3 2 RECURRENCE EXPRESSION OF THE MASS CENTER VECTORS AND THEIR DERIVATIONS WITH RESPECT TO TIME

We consider the two connected bodies in system as shown in Fig 3, where the deformed state (shadowed) and undeformed state are presented.

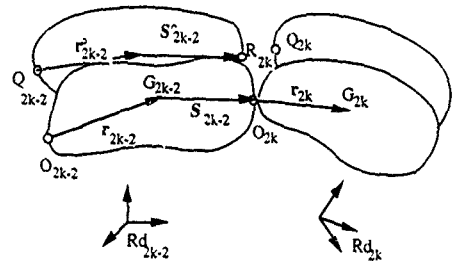


Fig. 3. The position of the mass center.

The position of the mass center of the body is

$$a_{2k} = a_{2k-2} + S_{2k-2} + r_{2k}, \tag{37}$$

in which

$$S_{2k-2} = S^{\circ}_{2k-2} + \xi_{(2k-2),\alpha}\lambda_{(2k-2),\alpha}$$

$$r_{2k} = r^{\circ}_{2k} + \xi_{2k,\alpha}\lambda_{2k,\alpha}$$

$$\lambda_{(2k-2),\alpha} = \Phi_{(2k-2),\alpha}(R_{2k-2}) - \Phi_{(2k-2),\alpha}(G_{2k-2})$$

$$\lambda_{2k,\alpha} = \Phi_{2k,\alpha}(G_{2k}) - \Phi_{2k,\alpha}(Q_{2k}).$$

r° and r are the vectors from a connecting point to a mass center, and, S° and S are the vectors from a mass center to a connecting point, respectively before and after the deformation of the body. To have a rapid computing of the time derivations of a, we define the following operators

$$D_f^k(p) = \omega_k \Lambda (\ ) \tag{38}$$

$$D_2^k(p,\dot{p}) = \omega_k \Lambda (\ ) + \omega_k \Lambda (\omega_k \Lambda (\ )) \tag{39}$$

$$D_3^k(p,p,p) = \omega_k \Lambda (\cdot) + 2\,\omega_k \Lambda (\omega_k \Lambda (\ )) + \omega_k \Lambda (\omega_k \Lambda (\ )) + \omega_k \Lambda (\omega_k \Lambda (\omega_k \Lambda (\ ))) \tag{40}$$

The time derivations of a using the operators are:

$$a_{2k} = a_{2k-2} + D_1^{2k-2}(p)s_{2k-2} + \xi_{(2k-1),\alpha}\,\lambda_{(2k-2),\alpha} + D_1^{2k}(\dot{p})r_{2k} + \xi_{2k,\alpha}\,\lambda_{2k,\alpha} \tag{41}$$

$$\dot{a}_{2k} = a_{2k-2} + D_2^{2k-2}(\dot{p},p)s_{2k-2} + 2D_1^{2k-2}(p)(\xi_{(2k-2),\alpha}\,\lambda_{(2k-2),\alpha})$$
$$+\xi_{(2k-2),\alpha}\,\lambda_{(2k-2),\alpha} + D_2^{2k}(p,\dot{p})r_{2k} + 2D_1^{2k}(\dot{p})(\xi_{2k,\alpha}\,\lambda_{2k,\alpha}) + \xi_{2k,\alpha}\,\lambda_{2k,\alpha} \tag{42}$$

$$\ddot{a}_{2k} = \ddot{a}_{2k-2} + D_3^{2k-2}(p,p,\dot{p})s_{2k-2} + 3D_2^{2k-2}(p,\ddot{p})(\xi_{(2k-2),\alpha}\,\lambda_{(2k-2),\alpha})$$
$$+ 3D_1^{2k-2}(\dot{p})(\dot{\xi}_{(2k-2),\alpha}\,\lambda_{(2k-2),\alpha}) + \ddot{\xi}_{(2k-2),\alpha}\,\lambda_{(2k-2),\alpha} + D_3^{2k}(p,p,\ddot{p})r_{2k}$$
$$+ 3D_2^{2k}(p,\dot{p})(\xi_{2k,\alpha}\,\lambda_{2k,\alpha}) + 3D_1^{2k}(\dot{p})(\dot{\xi}_{2k,\alpha}\,\lambda_{2k,\alpha}) + \xi_{2k,\alpha}\,\lambda_{2k,\alpha}. \tag{43}$$

We have obtained the recuurence expressions and the operators to calculate the high order time derivations of the angular velocities and of the positions of the mass center. The derivations of the mechanical tensor terms with respect to the generalized coordinates are given in Appendix. In the following, we give a brief about our programming

4. Sequence of Computation

4.1 DEFINATION OF THE POSITION FUNCTION

The following relationships allow to find the position of the corresponding coefficient vector in one dimension vector array for the given subscripts (e.g., i and j, or i, j, and k). We call these relationships the position functions:

a) For p_ip_j, $\quad LOC1(i,j) = \frac{1}{2}(i-1)i + j \qquad (j \geq i)$

b) For $p_ip_jp_k$, $\quad LOC2(i,j,k) = \frac{1}{6}(k-1)k(k+1) + \frac{1}{2}(i-1)i + j \qquad (k \geq i \geq j)$

c) For p_ip_j, $\quad LOC3\_1(i,j) = j(j-2) + 2i \qquad (i \leq j)$

$\qquad\qquad LOC3\_2(i,j) = i(i-2) + 2j + 1 \qquad (i > j)$

Relations between the positions of coefficients and the combinations of the time derivatives of the generalized coordinates are given in Table I.

Table I Correspondences between the positions of coefficients and the combinations of the time derivatives of the generalized coordinates

a	p_j	$\dot{p}_1$	$\dot{p}_2$	$\dot{p}_3$	p_4	....	p_l	Position function
(ω)	$A_i(p)$	a_1	a_2	a_3	a_4		a_l	
	position	*1*	*2*	*3*	*4*	....	*l*	
	$p_i\dot{p}_j$	$p_1\dot{p}_1$	p_2p_1	p_2p_2	$\dot{p}_3p_1$	. .	$\dot{p}_l\dot{p}_l$	
	$B_{ij}(p)$	B_{11}	B_{21}	B_{22}	B_{31}	.. .	B_{ll}	
a	position	*1*	*2*	*3*	*4*	.....	*l*	LOC1
(ω)	$\dot{p}_j$	p_1	$\dot{p}_2$	$\dot{p}_3$	p_4	.. ...	p_l	
	$C_i(p)$	C_1	C_2	C_3	C_4	... .	C_l	
	position	*1*	*2*	*3*	*4*	.....	*l*	
	$p_ip_j\dot{p}_k$	$p_1p_1p_1$	$\dot{p}_2p_1p_1$	$p_2p_2p_1$	$p_1p_1p_1$	.....	$p_l\dot{p}_lp_l$	
	$D_{ij}k(p)$	D_{111}	D_{211}	D_{221}	D_{222}	... .	D_{lll}	
	position	*1*	*2*	*3*	*4*	.....	*l*	LOC2
$\ddot{a}$	$\dot{p}_i\dot{p}_j$	$\ddot{p}_1p_1$	p_2p_1	$\dot{p}_1\dot{p}_2$	p_2p_2	. ...	$p_l\dot{p}_l$	
($\dot{\omega}$)	$E_{ij}(p)$	E_{11}	E_{21}	E_{12}	E_{22}	.....	E_{ll}	
	position	*1*	*2*	*3*	*4*	.....	*l*	I OC3_1 and LOC3_2
	$\dddot{p}_j$	$\dot{p}_1$	$\ddot{p}_2$	p_3	$\dot{p}_4$	. ..	p_l	
	$F_i(p)$	F_1	F_2	F_3	F_4	.....	F_l	
	position	*1*	*2*	*3*	*4*	...	*l*	

4 2 CALCULATIONS OF THE OPERATORS

In order to save the calculation time, we successively apply the recurrence expressions to the terms in the operators.

$$A_i^k = \omega_k \Lambda (\omega_k \Lambda e_i) = A_i^{k-1} + \bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda e_i) + \omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i) + \bar{U}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i)$$

$$B_i^k = \omega_k \Lambda (\omega_k \Lambda e_i) = B_i^{k-1} + (\omega_{k-1} \Lambda \bar{U}_{k-1}) \Lambda (\omega_{k-1} \Lambda e_i) + \bar{\bar{U}}_{k-1} \Lambda (\omega_{k-1} \Lambda e_i)$$

$$+ \omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i) + (\omega_{k-1} \Lambda \bar{U}_{k-1}) \Lambda (\bar{U}_{k-1} \Lambda e_i) + \bar{\bar{U}}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i)$$

$$C_i^k = \omega_k \Lambda (\omega_k \Lambda e_i) = C_i^{k-1} + \bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda e_i) + \omega_{k-1} \Lambda ((\omega_{k-1} \Lambda \bar{U}_{k-1}) \Lambda e_i)$$

$$+ \omega_{k-1} \Lambda (\ddot{\bar{U}}_{k-1} \Lambda e_i) + \bar{U}_{k-1} \Lambda ((\omega_{k-1} \Lambda \dot{\bar{U}}_{k-1}) \Lambda e_i) + \dot{\bar{U}}_{k-1} \Lambda (\dot{\bar{U}}_{k-1} \Lambda e_i)$$

$$D_i^k = \omega_k \Lambda (\omega_k \Lambda (\omega_k \Lambda e_i)) = D_i^{k-1}$$

$$+ \bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda (\omega_{k-1} \Lambda e_i)) + \omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda e_i))$$

$$+ \omega_{k-1} \Lambda (\omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i)) + \bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i))$$

$$+ \bar{U}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda (\omega_{k-1} \Lambda e_i)) + \omega_{k-1} \Lambda (\bar{U}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i))$$

$$+ \bar{U}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda (\bar{U}_{k-1} \Lambda e_i)),$$

in which

$\bar{U}_{k-1} = p_{k,\alpha}\, \theta_{k,\alpha}$ and $\dot{\bar{U}}_{k-1} = \dot{p}_{k,\alpha}\, \theta_{k,\alpha}$.

Based on the recurrence formulae as given earlier, the procedure for the numerical evaluations of the damping and stiffness matrices for the linearized system can be summerized as follows:

1) For each time step, calculating the vectors $\theta_1, \theta_2, \ldots, \theta_3$ (See the Eq.(26)),

2) Calculating the coefficients $\bar{A}_i$, $\bar{B}_{ij}$, $\bar{C}_i$, $\bar{D}_{ijk}$, and $\bar{F}_i$ in Eqn (23 to 25);

3) Calculating the operators $D_1^k(p)$, $D_2^k(\dot{p}, p)$, and $D_3^k(\dot{p}, p, \dot{p})$ (Eqn. (38 to 40));

4) Calculating the coefficients A_i, B_{ij}, C_i, D_{ijk}, and Γ_i in Eqn (19 to 21);

5) Calculating the partial p and $\dot{p}$ derivatives of kinematic (Eqn.(26 to 33))and mechanical terms (Appendix);

6) Finally, establishing the damping and stiffness matrices.

Figure 4 shows the flow chart of computational algorithm for the linearization procedure

5. Numerical Example

In order to demonstrate the performance of the numerical linearization method proposed in this paper, the program has been applied to a two rigid manipulator model, as shown in Fig 5. The numerical data of this mechanism, which are input data, are given in Table II Table III compares the analytical results through a direct partial derivation with those

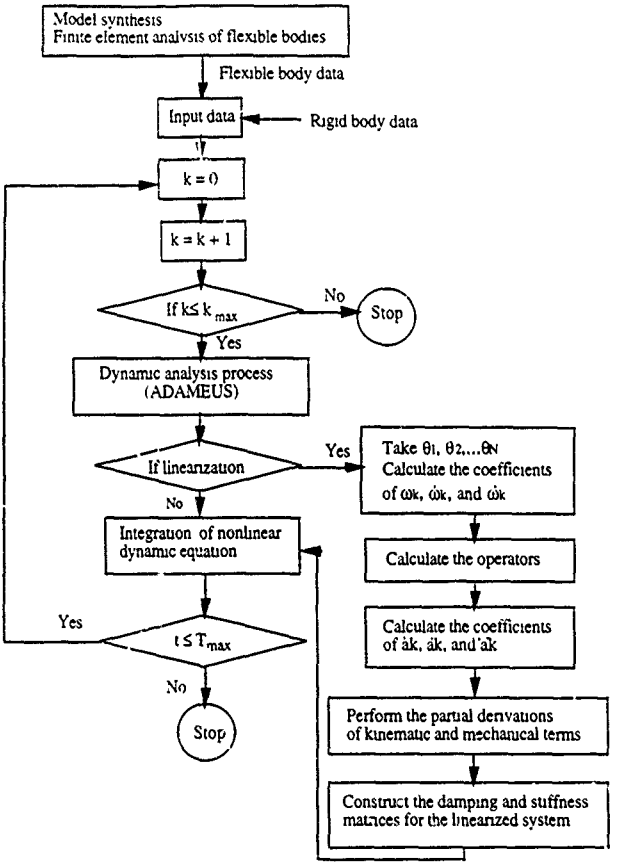


Fig 4 Computational algorithm

obtained by our numerical linearization procedure. They are in good agreement. The differences are possibly due to the inaccuracy of the hand calculation

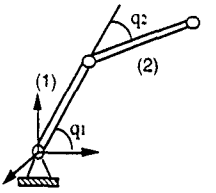


Fig.5 Two rigid planner manipulator model.

Table II Numerical data for the two links in the model

		Link (1) and Link(2)
Link length (m)		1
Link mass (kg)		60
Area moment of inertia	Ixx	5 037
(m^4)	Iyy	0.0733
	Izz	5.037

Table III Comparison of the analytical results with the numerical ones

	Analytical results		Numerical results	
Damping matrix	66 30	10 78	66 32	10 79
	77 08	0.00	76 91	0 00
Suffness matrix	516.88	520.12	513 92	516 98
	200.81	526.49	199 77	523 76

6. Conclusion

We have proposed a linearization procedure for the nonlinear dynamic equations, which are established in our case based on the Kane's virtual work principle. The partial derivations of the kinematic terms with respect to the generalized coordinates and to the generalized velocities have been fulfilled through the first and second order time derivations of the body absolute angular velocities and through the first, second, and third order time derivations of the mass center vectors, respectively We have developped the specific operators to perform the time derivations without distinguishing the rigid and

flexible variables. Thus, this method makes both the theoretical deduction and the programming relatively simple, and allows a rapid computing.

ACKNOWLEDGEMENT The authors wish to thank Drs G. Gallay and P Chapuis for their helps This work is supported by Societe Bertin et Cie, FRANCE

APPENDIX

We give the derivation of the mechanical tensor term with repect to the generalized coordinates

1 Definition of the mechanical tensor term

Mechanical tensor terms for the deformed body in the reference frame R_0 are defined as:

$$J_{ij} = \int [\mathbf{X} \wedge (\mathbf{e}_i \wedge \mathbf{X})] \cdot \mathbf{e}_j dm,$$

$$K_{ij} = \int [\mathbf{X} \wedge (\mathbf{e}_i \wedge \dot{\mathbf{X}})] \cdot \mathbf{e}_j dm,$$

$$J\alpha_{ij} = \int [\Delta\Phi_\alpha \wedge (\mathbf{e}_i \wedge \mathbf{X})] \cdot \mathbf{e}_j dm,$$

$$H_\alpha = \int [\mathbf{X} \wedge \Delta\Phi_\alpha] dm, \quad \dot{H}_\alpha = \int [\dot{\mathbf{X}} \wedge \Delta\Phi_\alpha] dm, \text{ and } \quad \mu_\alpha = \int \Delta\Phi_\alpha^2 \, dm,$$

in which

$$\Delta\Phi_\alpha = \Phi_\alpha(M) - \Phi_\alpha(G),$$

and X is a vetor on the deformed body, which is equal to $\mathbf{X} = \mathbf{X}_r + \xi_\alpha \Delta\Phi_\alpha$

2 Derivation of the vectors $\mathbf{X}$, $\dot{\mathbf{X}}$ and Φ_α with respect to the generalized coordinates

$$\frac{\partial \mathbf{X}}{\partial p_k} = \theta_k \wedge \mathbf{X}, \qquad \frac{\partial \dot{\mathbf{X}}}{\partial p_k} = \theta_k \wedge \dot{\mathbf{X}}, \qquad \frac{\partial \mathbf{x}}{\partial \xi_\alpha} = \Delta\Phi_\alpha, \qquad \text{and} \qquad \frac{\partial \Delta\Phi_\alpha}{\partial p_k} = \theta_k \wedge \Delta\Phi_\alpha$$

3 Derivation of the mechanical tensor terms with respect to the generalized coordinates

a. $$\frac{\partial J_{ij}}{\partial p_k} = \int \left[\frac{\partial \mathbf{x}}{\partial p_k} (\mathbf{e}_i \wedge \mathbf{X}) \cdot \mathbf{e}_j dm + \int [\mathbf{X} \wedge (\mathbf{e}_i \wedge \frac{\partial \mathbf{x}}{\partial p_k})] \cdot \mathbf{e}_j dm\right.$$

$$= \theta_{km} e_m \cdot \{ \int -[(e_i \cdot X)(X \Lambda e_j) + (X e_j)(X \Lambda e_i)] dm \}$$

$$\begin{bmatrix} \frac{\partial J_{11}}{\partial p_k} & \frac{\partial J_{21}}{\partial p_k} & \frac{\partial J_{31}}{\partial p_k} \\ \frac{\partial J_{12}}{\partial p_k} & \frac{\partial J_{22}}{\partial p_k} & \frac{\partial J_{32}}{\partial p_k} \\ \frac{\partial J_{13}}{\partial p_k} & \frac{\partial J_{23}}{\partial p_k} & \frac{\partial J_{33}}{\partial p_k} \end{bmatrix} =$$

$$\begin{bmatrix} 2J_{13}\theta_{k2} - 2J_{12}\theta_{k3} & -J_{13}\theta_{k1} + J_{23}\theta_{k2} + (J_{11} - J_{22})\theta_{k3} & J_{12}\theta_{k1} + (J_{33} - J_{11})\theta_{k2} - J_{23}\theta_{k3} \\ -J_{13}\theta_{k1} + J_{23}\theta_{k2} + (J_{11} - J_{22})\theta_{k3} & -2J_{23}\theta_{k1} + 2J_{12}\theta_{k3} & (J_{22} - J_{33})\theta_{k1} - J_{12}\theta_{k2} + J_{13}\theta_{k3} \\ J_{12}\theta_{k1} + (J_{33} - J_{11})\theta_{k2} - J_{23}\theta_{k3} & (J_{22} - J_{33})\theta_{k1} - J_{12}\theta_{k2} + J_{13}\theta_{k3} & 2J_{23}\theta_{k1} - 2J_{13}\theta_{k2} \end{bmatrix}$$

$$\frac{\partial J_{ij}}{\partial \xi_\alpha} = \int [\frac{\partial x}{\partial \xi_\alpha} (e_i \Lambda X) e_j] dm + \int [X \Lambda (e_i \Lambda \frac{\partial x}{\partial \xi_{j\alpha}}] e_j dm = J_\alpha + J_\alpha^T$$

b $$\frac{\partial k_{ij}}{\partial p_k} = \int [\frac{\partial x}{\partial p_k} (e_i \Lambda \hat{X}) \cdot e_j dm + \int [X \Lambda (e_i \Lambda \frac{\partial \hat{X}}{\partial p_k})] \cdot e_j dm$$

$$= \theta_{km} e_m \cdot \{ \int -[(X \Lambda \hat{X})(e_i e_j) - (X \Lambda e_i)(\hat{X} e_j) - (\hat{X} e_i)(X \Lambda e_j) - (X \hat{X} e_j) e_i] dm \}$$

$$\begin{bmatrix} \frac{\partial k_{11}}{\partial p_k} & \frac{\partial k_{21}}{\partial p_k} & \frac{\partial k_{31}}{\partial p_k} \\ \frac{\partial k_{12}}{\partial p_k} & \frac{\partial k_{22}}{\partial p_k} & \frac{\partial k_{32}}{\partial p_k} \\ \frac{\partial k_{13}}{\partial p_k} & \frac{\partial k_{23}}{\partial p_k} & \frac{\partial k_{33}}{\partial p_k} \end{bmatrix} =$$

$$\begin{bmatrix} (k_{31}+k_{13})\theta_{k2}-(k_{12}+k_{21})\theta_{k3} & -k_{31}\theta_{k1}+k_{23}\theta_{k2}+(k_{11}-k_{22})\theta_{k3} & k_{21}\theta_{k1}+(k_{33}-k_{11})\theta_{k2}-k_{32}\theta_{k3} \\ -k_{13}\theta_{k1}+k_{32}\theta_{k2}+(k_{11}-k_{22})\theta_{k3} & -(k_{23}+k_{32})\theta_{k1}+(k_{12}+k_{21})\theta_{k3} & (k_{22}-k_{33})\theta_{k1}-k_{12}\theta_{k2}+k_{31}\theta_{k3} \\ k_{12}\theta_{k1}+(k_{33}-k_{11})\theta_{k2}-k_{23}\theta_{k3} & (k_{22}-k_{33})\theta_{k1}-k_{21}\theta_{k2}+k_{13}\theta_{k3} & (k_{23}+k_{32})\theta_{k1}-(k_{13}+k_{31})\theta_{k2} \end{bmatrix}$$

$$\frac{\partial k_{ij}}{\partial \xi_\alpha} = \int \left[\frac{\partial x}{\partial \xi_\alpha} \ (e_i \wedge \hat{x})\, e_j dm \ = \int [\ \Delta\Phi_\alpha \wedge (e_i \wedge \hat{x})]\ e_j\, dm\right.$$

c $$\frac{\partial H_\alpha}{\partial p_k} = \int \frac{\partial x}{\partial p_k} \wedge \Delta\Phi_\alpha\, dm + \int x \wedge \frac{\partial \Delta\Phi_\alpha}{\partial p_k} dm = \theta_k \wedge H_\alpha$$

d. $$\frac{\partial \hat{H}_\alpha}{\partial p_k} = \int \frac{\partial x}{\partial p_k} \wedge \Delta\Phi_\alpha\, dm + \int x \wedge \frac{\partial \Delta\Phi_\alpha}{\partial p_k} dm = \theta_k \wedge \hat{H}_\alpha$$

Note that we did not give the derivation of $j_{\alpha ij}$ since it is similar to K_{ij} in form

REFERENCES

[1] C W CHANG Gartesian-based control of a deformable robot manipulator, computers & structures Vol 34, No.4, pp.527-536 (1990).

[2] J J CRAIG, Adaptive control of mechanical manipulator, Addison-Wesley, Reading MA (1987).

[3] G GALLAY, Commande dynamique de mecanismes deformables Document Bertin N°87, la 06.

[4] J T WANG and R.L.HUSTON, Kane's equation witch undetermined multipliers —application to constrained multibody systems Journal of applied mecanics Vol.54, June (1987).

[5] T. R KANE, Dynamics of nonholonomic system, J. Appl of Mechanics, Vol. 28, Trans ASME, Vol. 83, series E, Dec pp 574 (1961).

EFFICIENT MODELLING OF GENERAL MULTIBODY DYNAMIC SYSTEMS WITH FLEXIBLE COMPONENTS

Kurt S Anderson
Institut für Mechanik, Technische Hochschule Darmstadt
Hochschulstrasse 1
64289 Darmstadt, Germany

ABSTRACT. This paper presents a computer based method for the efficient formulation and solution of the nonlinear equations of motion for mechanical systems of interconnected flexible bodies, subject to motion and/or geometric constraints, undergoing elastic deformation together with large rotations and translations. Flexibility is modeled through the use of admissible functions obtained from previous finite element analysis of the component bodies. The procedure then forms the equations of motion and solves for the system state derivatives associated with the unconstrained system in a highly efficient *Order n* manner The constraint loads required to enforce the constraint relations are subsequently determined through the use of a constraint stabilization method The required constraint loads are then used to modify the state derivatives found previously, resulting in a set of state derivative values which are now associated with the constrained system. The procedure is efficient, accounts for motion induced stiffness, and produces equations which are highly concurrent in form, yielding simulation code which lends itself well to application on parallel computers

1. Introduction

Much work has been done in the development of simulation procedures and computer codes for multibody dynamic systems since the pioneering work of Hooker and Margulies [1] over twenty-five years ago. During this period many simulation procedures and associated simulation codes have been developed for treating a wide variety of systems. When the systems are modeled as containing only rigid bodies, the resulting equations of motion are exact However, in many applications some or all of the component bodies must be treated as flexible, and in these instances the story is quite different Here, one finds many assumptions, approximations, computational concerns, open issues, and a good deal of debate

Complicating the situation still further is the issue of computational speed and

throughput. In the past it was often the case that computational efficiency and speed were sacrificed for the sake of codes which could handle more general systems, or that accuracy was often sacrificed for the sake of computational speed [2] However, for analytical modeling and computer simulation to be an effective tool, it must be accurate, fast, and economical The design process may require many simulations, and may thus be limited by time and/or monetary constraints As a result, recent years have seen great emphasis placed on developing algorithms and simulation programs which are computationally efficient/economical, while remaining general enough to be adequately used as analysis tools for a wide variety of rigid and flexible multibody systems [2-9].

Presented in this paper is one such approach, based on a highly efficient *Order n* algorithm, for dealing with multibody dynamic systems. The recursive method uses relative coordinates for rigid body displacements, and shape functions obtained from finite element modal analysis of the component bodies as admissible functions for the representation of elastic body flexibility The formulation considers motion induced stiffening (too often neglected), and the approach can be applied to both tree and closed loop configurations. The resulting equations of motion are highly parallel in form and lend themselves well to application on loosely coupled distributed architecture parallel computers

2. Analytical Development

Consider a system of $\mathcal{N}$ interconnected flexible bodies The flexibility of each body is approximated through the use of a finite set of admissible shape functions and associated modal coordinates, with the interbody joints each contributing from zero to six rigid body degrees of freedom to the system total. If the total number flexible body and rigid body degrees of freedom is n, and P is define to be a differential element of an arbitrary body of the system, then the angular velocity and velocity of P in the inertia reference frame $\mathcal{N}$ can be respectively expressed as

$$\boldsymbol{\omega}^P = \sum_{r=1}^{n} \boldsymbol{\omega}_r^P u_r + \boldsymbol{\omega}_t^P \qquad \text{and} \qquad \mathbf{v}^P = \sum_{r=1}^{n} \mathbf{v}_r^P u_r + \mathbf{v}_t^P, \tag{1}$$

where, by definition [10]: $u_1, \ldots, u_n$ are *generalized speeds*, quantities which characterize the motion of the system, and, $\boldsymbol{\omega}_r^P$ and $\mathbf{v}_r^P$ are the rth partial angular velocity and partial velocity of P in $\mathcal{N}$, respectively

If $\mathbf{R}$ represents the resultant of all distance, contact, and elastic forces acting on P, and $\mathbf{R}^*$ are the inertia forces associated with P, then the equations of motion are given by

$$\sum_{k=1}^{\mathcal{N}} \int_{V^k} \mathbf{v}_r \cdot \mathbf{R} dV^k + \sum_{k=1}^{\mathcal{N}} \int_{V^k} \mathbf{v}_r \cdot \mathbf{R}^* dV^k = 0 \qquad (r = 1, \ldots, n) \tag{2}$$

Unfortuately, proceeding in this manner yields a number of difficulties The equations of motion which are produced as per Eqs (2) are incorrectly linearized in the modal

coordinates and there time derivatives derivatives This can lead to significant error where motion induced stiffness is involved. In addition, the integration over the volume of each body and the indicated summations over all bodies of the system at each time step can be computationally expensive Furthermore, the resulting equations of motions are highly coupled in the state derivatives, thus requiring computationally expensive decomposition of the mass matrix before the state derivatives can be solved for and the equations of motion can be temporally integrated Each of these difficulties will be addressed subsequently.

2.1 NOTATION AND GEOMETRY

Once the identification number k for each body in the system has been properly assigned, the topology of the system is uniquely defined by identification numbers of the proximal bodies The notation $Pr[k]$ refers to the set containing only the proximal body of the B^k. Pr is called the *proximal body array*, and the convention that $Pr[k] = 0$ is used if k is a base body where bodies 0 and N are synonymous labels for the inertial reference frame It is also useful to introduce an additional set $Dist[k]$ to characterize the system's topology. The notation $Dist[k]$ refers to a set of body numbers defined as

$$Dist[k \triangleq \{j \mid Pr[j] = k\} \tag{3}$$

where $Dist$ is called the *distal body set array*

Throughout this paper, scalar quantities will be represented as plain faced Vector quantities will be denoted by bold symbols, while matrices are shown as plain faced with an underbar. Dyadics are represented by bold faced symbols with an under tilde, while matrices composed of either vector or dyadic quantities are represented by bold faced symbols with an underbar. For example, the symbols A, $\mathbf{A}$, $\underline{\mathbf{d}}$, $\underline{\mathbf{A}}$, and $\underline{\mathbf{A}}$, represent a scalar, vector, matrix, dyadic, and matrix of vector or dyadic quantities, respectively Unless stated otherwise, a variable in the subscript preceded by a comma indicates differentiation with respect to that variable, and summations are carried out over repeated indices, with indices $l, m, o, p = 1, 2, 3, q = 1, \quad, 21$, $r, s, t = 1, \quad, \mu^{B^k}$, and $j = 1, \quad, \mu^{J^k}$. For each of these, μ^{B^k} and μ^{J^k} are, respectively, the number of flexible degrees of freedom associated with body B^k, and the number of rigid body degrees of freedom associated with joint J^k connecting B^k to $B^{Pr[k]}$

The joints between adjacent bodies can be thought of as being composed of a series of properly oriented single degree of freedom revolute or prismatic "subjoints" connected via massless/dimensionless links. Thus, if joint J^k is a six degree of freedom *free joint*, then it can be described by the set of single degree of freedom subjoints $J_1^k, \quad, J_6^k$. The value of generalized coordinate $q^{J_l^k}$ representing the relative position/orientation of the subjoint/body J_l^k with respect to its proximal subjoint/body J_{l-1}^k in the direction of unit vector $\boldsymbol{\gamma}_l^k$

A typical flexible body of the system is defined as a body B^k which undergoes relative rigid body motion with respect to its proximal body $B^{Pr[k]}$ and deforms elastically, Fig 1 Let $\mathbf{n}^{B^k J^k}$ be the reference frame of B_k with respect to which the

deformation of the body is given. Reference frame $\mathbf{n}^{B^k,i}$ represents the frame within which the dextral mutually perpendicular unit vectors $\mathbf{n}_1^{B^k,i}$, $\mathbf{n}_2^{B^k,i}$, $\mathbf{n}_3^{B^k,i}$ are fixed and which is itself fixed with respect to point $P^{B^k,i}$, the i^{th} grid point of the finite element representation of B^k. The point for which $i = J^k$, is that point of B^k which connects B^k to $B^{P[k]}$ through joint J^k.

The position of an arbitrary $P^{B^k,i}$ of B^k with respect to P^{k,J^k} is given by the position vector

$$\boldsymbol{\xi}^{k,i} \triangleq \boldsymbol{\rho}^{k,i} + \boldsymbol{\phi}_s^{k,i} q_s^{B^k} \tag{4}$$

where $\boldsymbol{\rho}^{k,i}$ and $\underline{\boldsymbol{\phi}}^{k,i}$, are the position vector of $P^{k,i}$ in undeformed B^k, and the admissible shape function matrix associated with the translation of $P^{k,i}$, respectively. The body B^k has μ^{B^k} modal deformation coordinates associated with it, where $q_s^{B^k}$ is the s^{th} coordinate. In a like manner, the angular position of a node at $P^{k,i}$ with respect to P^{J^k}, $\theta^{k,i}$, arising from the deformation of B^k, is expressed in terms of modal coordinates, $q_s^{B^k}$, and rotational deformation shape functions. If $\psi_s^{k,i}$ is defined to be the rotational shape function associated with $q_s^{B^k}$ at $P^{k,i}$ with respect to P^{J^k}, then $\theta^{k,i}$ is given by

$$\theta_l^{k,i} \triangleq \psi_{ls}^{k,i} q_s^{B^k} \tag{5}$$

Finally, unless otherwise specified, all time differentiations are taken in the inertia frame.

2.2 MODAL REPRESENTATION AND GEOMETRIC STIFFENING

To reduce the large number of elastic coordinates, a standard component mode technique is utilized [11]. This approach involves using a relatively small number of shape functions for each elastic body. The shape functions used consist of selected vibration modes, all constraint modes, and any necessary static correction modes. The important issue of how these modes are best selected is a area much on going work in the multibody dynamics community and will not be addressed here.

Linear strain energy theory assumes that the deformation components are independent. But in systems involving *high* rotation rates, high radial forces can occur and the coupling between radial and transverse deflections becomes significant. Unfortunately, the use of modal coordinates in modeling flexibility results in equations of motion which are incorrectly linearized with respect to these coordinates and there time derivatives. For this reason higher order strain energy terms need to be considered. These terms are used in the generation of geometric stiffness expressions which are added to the existing equations to correct for the premature linearization in the modal terms. Methods such as those discussed in references [12] and [13] produce geometric stiffness terms which require iterative updates, and thus are computationally expensive. By comparison, reference [14] presents a technique which applies to general flexible structures and requires the determination of a set of geometric stiffness matrices only once.

This latter technique is the one which is used in this paper, modified to make this method compatible with the $O(n)$ formulation As given in Ref. [14], the element geometric stiffness is

$$\underline{k}_g^e = \int_e \begin{bmatrix} \underline{N}_{,x}^T & \underline{N}_{,y}^T & \underline{N}_{,z}^T \end{bmatrix} \begin{bmatrix} \sigma_{xx0}\underline{U} & \sigma_{xy0}\underline{U} & \sigma_{xz0}\underline{U} \\ \sigma_{xy0}\underline{U} & \sigma_{yy0}\underline{U} & \sigma_{yz0}\underline{U} \\ \sigma_{xz0}\underline{U} & \sigma_{yz0}\underline{U} & \sigma_{zz0}\underline{U} \end{bmatrix} \begin{bmatrix} \underline{N}_{,x} \\ \underline{N}_{,y} \\ \underline{N}_{,z} \end{bmatrix} dV, \tag{6}$$

where $\underline{N}(x,y,z)$ is the $(3 \times e\text{-}ndof)$ matrix of interpolation functions, with x, y, z, as local coordinates. The stresses σ_{ij0} $(i,j = x,y,z)$ are those arising from the loads applied to the nodes of the element, the quantity $e\text{-}ndof$ represents the number of element degrees-of-freedom, and $\underline{U}$ is a (3×3) identity matrix

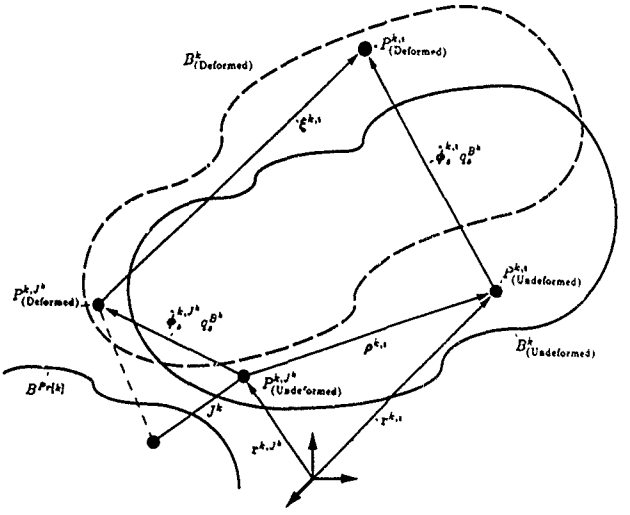


Fig 1. Notation for describing flexible body

Finite element codes, such as NASTRAN can compute the geometric stiffness associated with a prescribed distributed load by first calculating the associated element stresses, these element stresses then being used in the determination of the element geometric stiffness matrices as per Eq.(6) Finally, the element geometric stiffness

matrices are assembled into the body geometric stiffness matrix $\underline{K}_G^{B^k}$

The overall geometric stiffness matrix for B^k, $\underline{K}_G^{B^k}$, is constructed from twenty-one time invariant contributing matrices, $\underline{\hat{K}}_{G_q}^{B^k}$ $(q = 1, \ldots, 21)$, and their associated temporal scalars, $A_{G_q}^{B^k}$ which arise from the inertia loads in B^k. In addition, there are another six contributing time invariant matrices, $\underline{\hat{K}}_{G_{J_h}}^{B^k}$ $(J \in Dist[k],\ h = 1, \ldots, 6)$, and their associated temporal scalars, $F_h^{J^j}$, which are affiliated with each distal body to B^k and account for the geometric stiffening in B^k due to loads applied to it by each of these distal bodies through the corresponding interbody joint, J^j. Specifically,

$$\underline{K}_G^{B^k} = \underline{\hat{K}}_{G_q}^{B^k}\left(\hat{A}_{G_q}^{B^k} + \bar{A}_{G_q}^{B^k}\right) + \sum_{J \in Dist[k]} \underline{\hat{K}}_{G_{J_h}}^{B^k} F_h^{J^j} \tag{7}$$
$$(q = 1, \ldots, 21;\ J \in Dist[k];\ h = 1, \ldots, 6)$$

2.3 UNCONSTRAINED SYSTEMS

Key to the reduction in the number of required operations is the removal of summations which appear in Eq (2) and the associated geometric stiffness corrective terms. The indicated summation can be eliminated entirely through the recursive shifting of active and inertia forces to their proximal bodies. The state derivatives can then be determined inexpensively through the recursive triangularization and back-substitution of the resulting equations [6-8]. The arguments of the integrals may be split into spatial and temporal parts, with the indicated spatial integrations needing to be carried out only once. The temporal quantities are then multiplied by time invariant coefficients resulting from the spatial integration.

Determination of Generalized Forces and Triangularization of Equations. Differentiating Eqs (1) with respect to time yields

$$\boldsymbol{\alpha}^P = \frac{{}^N d\boldsymbol{\omega}^P}{dt} = \sum_{r=1}^{n} \boldsymbol{\omega}_r^P u_r + \left(\sum_{r=1}^{n} \boldsymbol{\omega}_r^P u_r + \boldsymbol{\omega}_t^P\right) = \bar{\boldsymbol{\alpha}}^P + \hat{\boldsymbol{\alpha}}^P \tag{8}$$

and

$$\mathbf{a}^P = \frac{{}^N d\mathbf{v}^P}{dt} = \sum_{r=1}^{n} \mathbf{v}_r^P u_r + \left(\sum_{r=1}^{n} \mathbf{v}_r^P u_r + \mathbf{v}_t^P\right) = \bar{\mathbf{a}}^P + \hat{\mathbf{a}}^P \tag{9}$$

for the angular acceleration of P in N, and the acceleration of P in N, respectively. The quantities $\bar{\boldsymbol{\alpha}}^P$, $\bar{\mathbf{a}}^P$, contain all terms explicit in the unknown state derivatives, $u_1, \ldots, u_n$, while $\hat{\boldsymbol{\alpha}}^P$ and $\hat{\mathbf{a}}^P$ contains all else.

Now consider P not as an infinitesimal particle, but as a discrete quantity, either a grid point $P^{B^k_i}$ of a finite element discretization of the body, or a subjoint J_l^k. Then the mass and central inertia dyadics associated with P, $\underline{\mathbf{M}}^P$ and $\underline{\mathbf{I}}^P$, are defined as

$$\underline{\mathbf{M}}^P \triangleq M^P \mathbf{n}_l^P \mathbf{n}_l^P \quad \text{and} \quad \underline{\mathbf{I}}^P \triangleq I_{lm}^P \mathbf{n}_l^P \mathbf{n}_m^P \tag{10}$$

where M^P is the mass lumped at P

If $\boldsymbol{\rho}^P$ is written as $\boldsymbol{\rho}^P = \rho_1^P \mathbf{n}_1^P + \rho_2^P \mathbf{n}_2^P + \rho_3^P \mathbf{n}_3^P$, then we define the dyadic $\underset{\sim}{\rho^P}\times$ as

$$\underset{\sim}{\rho^P}\times \triangleq -\rho_3^P \mathbf{n}_1^P \mathbf{n}_2^P - \rho_2^P \mathbf{n}_1^P \mathbf{n}_3^P + \rho_3^P \mathbf{n}_2^P \mathbf{n}_1^P - \rho_1^P \mathbf{n}_2^P \mathbf{n}_3^P - \rho_2^P \mathbf{n}_3^P \mathbf{n}_1^P + \rho_1^P \mathbf{n}_3^P \mathbf{n}_2^P \quad (11)$$

Having the quanties $\bar{\boldsymbol{\alpha}}^P$, $\bar{\mathbf{a}}^P$, $\hat{\boldsymbol{\alpha}}^P$, $\hat{\mathbf{a}}^P$, $\underset{\sim}{M}^P$, $\underset{\sim}{I}^P$, and $\underset{\sim}{\rho^P}\times$ so defined, introduce $\underset{\sim}{\mathcal{P}}_r^{k,i}$, $\underset{\sim}{\boldsymbol{\Phi}}_r^{J_j^k}$, $\underset{\sim}{\mathcal{I}}_1$, $\underset{\sim}{\mathcal{F}}_1$, $\underset{\sim}{\bar{\mathcal{A}}}$, and $\underset{\sim}{\mathcal{S}}$ for notational ease as

$$\underset{\sim}{\mathcal{P}}_r^{J_j^k} \triangleq \begin{bmatrix} \boldsymbol{\omega}_r^{J_j^k} \\ \mathbf{v}_r^{J_j^k} \end{bmatrix}, \qquad \underset{\sim}{\boldsymbol{\Phi}}_r^{k,i} \triangleq \begin{bmatrix} \boldsymbol{\psi}_r^{k,i} \\ \boldsymbol{\phi}_r^{k,i} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\boldsymbol{\Phi}}^{k,i} \triangleq \left[\underset{\sim}{\boldsymbol{\Phi}}_1^{k,i}, \ \ldots, \underset{\sim}{\boldsymbol{\Phi}}_{\mu^{B^k}}^{k,i} \right], \quad (12)$$

$$\underset{\sim}{\mathcal{I}}_1^{B^k,i} \triangleq \begin{bmatrix} \underset{\sim}{I}^{B^k,i} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{M}^{B^k,i} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\mathcal{I}}_1^{J_j^k} \triangleq \begin{bmatrix} \underset{\sim}{I}^{J_j^k} & \underset{\sim}{0} \\ \underset{\sim}{0} & \underset{\sim}{M}^{J_j^k} \end{bmatrix}. \quad (13)$$

$$\underset{\sim}{\mathcal{F}}_1^{B^k,i} \triangleq \begin{bmatrix} -\underset{\sim}{I}^{k,i} \cdot \hat{\boldsymbol{\alpha}}^{B^k,i} - \boldsymbol{\omega}^{B^k,i} \times \underset{\sim}{I}^{B^k,i} \ \boldsymbol{\omega}^{B^k,i} + \mathbf{T}^{B^k,i} \\ -\underset{\sim}{M}^{B^k,i} \ \hat{\mathbf{a}}^{B^k,i} + \mathbf{R}^{B^k,i} \end{bmatrix} \quad \text{and}$$

$$\underset{\sim}{\mathcal{F}}_1^{J_j^k} \triangleq \begin{bmatrix} -\underset{\sim}{I}^{J_j^k} \ \hat{\boldsymbol{\alpha}}^{J_j^k} - \boldsymbol{\omega}^{J_j^k} \times \underset{\sim}{I}^{J_j^k} \cdot \boldsymbol{\omega}^{J_j^k} + \mathbf{T}^{J_j^k} \\ -\underset{\sim}{M}^{J_j^k} \cdot \hat{\mathbf{a}}^{J_j^k} + \mathbf{R}^{J_j^k} \end{bmatrix}, \quad (14)$$

$$\underset{\sim}{\bar{\mathcal{A}}}^{B^k,i} \triangleq \begin{bmatrix} \bar{\boldsymbol{\alpha}}^{B^k,i} \\ \bar{\mathbf{a}}^{B^k,i} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\bar{\mathcal{A}}}^{J_j^k} \triangleq \begin{bmatrix} \bar{\boldsymbol{\alpha}}^{J_j^k} \\ \bar{\mathbf{a}}^{J_j^k} \end{bmatrix}, \quad (15)$$

$$\underset{\sim}{\mathcal{S}}^{B^k,i} \triangleq \begin{bmatrix} \underset{\sim}{U} & \underset{\sim}{\xi^{k,i}}\times \\ \underset{\sim}{0} & \underset{\sim}{U} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\mathcal{S}}^{J_j^k} \triangleq \begin{bmatrix} \underset{\sim}{U} & \underset{\sim}{\gamma_j^k}\times \\ \underset{\sim}{0} & \underset{\sim}{U} \end{bmatrix} \quad (16)$$

The terms $\mathbf{T}^P$ and $\mathbf{R}^P$ appearing in Eq (14) are the torque and force which line of action passes through P, and which together are equivalent to the set of all distance, contact and elastic forces (except those arising from geometric stiffness) acting on P In addition, from the corrective term matrix, $\underset{\sim}{\mathcal{R}_G^{B^k}}$, of forces/torques which arise due to geometric stiffness and given by

$$\underset{\sim}{\mathcal{R}_G^{B^k}} = \underset{\sim}{K_G^{B^k}} \underset{\sim}{\delta^{B^k}} = \underset{\sim}{K_G^{B^k}} \underset{\sim}{\Phi_r^{B^k}} q_r^{B^k}, \quad (17)$$

through matrix manipulation, one can construct

$$\underset{\sim}{\mathcal{F}_G}^{B^k,i} = \begin{bmatrix} \mathbf{T}_g^{B^k,i} \\ \mathbf{R}_g^{B^k,i} \end{bmatrix} = \left(\left[\underset{\sim}{\hat{\mathcal{I}}_t}^{B^k,i} \ \underset{\sim}{\bar{\mathcal{A}}}^{B^k,J^k} + \underset{\sim}{\hat{\mathcal{F}}_{G_t}}^{B^k,i} \right] + \sum_{j \in Dist[k]} \underset{\sim}{\hat{\mathcal{I}}_t}^{B_j^k,i} \left[\underset{\sim}{\mathcal{I}_3}^{J_1^j} \ \underset{\sim}{\bar{\mathcal{A}}}^{J_1^j} + \underset{\sim}{\mathcal{F}_3^{J_1^j}} + (\underset{\sim}{\mathcal{P}}_{J_1^j}^{J_1^j} \ \underset{\sim}{\mathcal{F}}^{J_1^j}) \underset{\sim}{\mathcal{P}}_{J_1^j}^{J_1^j} \right] \right) q_t^{B^k}, \quad (18)$$

where the vector quantities $\mathbf{T}_{\xi}^{B^k,i}$ and $\mathbf{R}_{\xi}^{B^k,i}$ are the torque/force pair which acts on grid i of B^k due to geometric stiffness: $\hat{\underline{T}}^{B^k,i}$ are matrices of dyadic quantities associated with the geometric stiffness due to body B^k inertia loads: $\hat{\underline{T}}^{B_h^k,i}$ are a matrices of dyadic quantities associated with the geometric stiffness due to applied constraint, and/or interbody loads acting at point h of B^k, and $\underline{\mathcal{F}}^{J_1^k} = [\mathbf{T}^{J_1^k}, \mathbf{R}^{J_1^k}]^T$

Starting from the terminal bodies and working inward, the composite inertia values as well as the active force and inertia remainder term portions of the inertia forces are determined for the triangularized equations by recursively using the relationships

$$\underline{\mathcal{F}}_3^{B^k,i} \triangleq \begin{cases} \underline{\mathcal{F}}_1^{B^k,i} & \text{if } i \neq J^{Dist[k]} \\ \underline{\mathcal{F}}_3^{J_0^{Dist[k]}} & \text{if } i = J^{Dist[k]} \end{cases} \tag{19}$$

$$\underline{\mathcal{I}}_3^{B^k,i} \triangleq \underline{\mathcal{I}}_1^{B^k,i} + \begin{cases} \underline{\mathcal{I}}_1^{B^k,i} & \text{if } i \neq J^{Dist[k]} \\ \underline{\mathcal{I}}_3^{J_0^{Dist[k]}} & \text{if } i = J^{Dist[k]} \end{cases} \tag{20}$$

Where $\underline{\mathcal{I}}_3^{B^k,J^k}$ and $\underline{\mathcal{F}}_3^{B^k,J^k}$ are defined as

$$\begin{aligned} \underline{\mathcal{I}}_3^{B^k,J^k} \triangleq \sum_{i}^{\nu^k} \Bigg(& \underline{\mathcal{S}}^{B^k,i} \cdot \underline{\mathcal{I}}_3^{B^k,i} \cdot (\underline{\mathcal{S}}^{B^k,i})^T \\ & + \left(\underline{\mathcal{S}}^{B^k,i} \cdot \underline{\mathcal{I}}_3^{B^k,i} \cdot \underline{\Phi}_s^{k,i} \right) \underline{g}_s^{B^k} + \underline{\mathcal{S}}^{B^k,i} \cdot \hat{\underline{\mathcal{I}}}_t^{B^k,i} q_t^{B^k} \\ & + \sum_{j \in Dist[k]} \left[\underline{\mathcal{S}}^{B^k,i} \cdot \hat{\underline{\mathcal{I}}}_t^{B_j^k,i} q_t^{B^k} \cdot \underline{\mathcal{I}}_3^{B^k,J_1^j} \cdot (\underline{\mathcal{S}}^{B^k,j} + \underline{\Phi}_s^{k,j} \cdot \underline{g}_s^{B^k}) \right] \Bigg), \end{aligned} \tag{21}$$

and

$$\begin{aligned} \underline{\mathcal{F}}_3^{B^k,J^k} \triangleq \sum_{i}^{\nu^k} \Bigg(& \underline{\mathcal{S}}^{B^k,i} \cdot \underline{\mathcal{F}}_3^{B^k,i} + \left(\underline{\mathcal{S}}^{B^k,i} \cdot \underline{\mathcal{I}}_3^{B^k,i} \cdot \underline{\Phi}_s^{k,i} \right) d_s^{B^k} + \underline{\mathcal{S}}^{B^k,i} \cdot \hat{\underline{\mathcal{F}}}_{G_t}^{B^k,i} q_t^{B^k} \\ & + \sum_{j \in Dist[k]} \left[\underline{\mathcal{S}}^{B^k,i} \cdot \hat{\underline{\mathcal{I}}}_t^{B_j^k,i} q_t^{B^k} \cdot (\underline{\mathcal{F}}_3^{J_1^j} + (\underline{\mathcal{P}}_{J_1^j}^{J_1^j} \cdot \underline{\mathcal{F}}^{J_1^j}) \underline{\mathcal{P}}_{J_1^j}^{J_1^j} + \underline{\mathcal{I}}_3^{J_1^j} \cdot \underline{\Phi}_s^{k,j} d_s^{B^k}) \right] \Bigg) \end{aligned} \tag{22}$$

with

$$\mathcal{M}_{rs}^{B^k} \triangleq \left[\sum_{i}^{\nu^k} (\underline{\Phi}_r^{k,i})^T \cdot \underline{\mathcal{I}}_3^{B^k,i} \cdot \underline{\Phi}_s^{k,i} \right] + \sum_{j \in Dist[k]} \left(\left[\sum_{i}^{\nu^k} (\underline{\Phi}_r^{k,i})^T \cdot \hat{\underline{\mathcal{I}}}_t^{B_j^k,i} \right] q_t^{B^k} \cdot \underline{\mathcal{I}}_3^{J_1^j} \cdot \underline{\Phi}_s^{B^k,j} \right), \tag{23}$$

and

$$\underline{d}^{B^k} = [d_1^{B^k}, \ldots, d_{\mu^{B^k}}^{B^k}]^T \triangleq (\underline{\mathcal{M}}^{B^k})^{-1} \underline{c}^{B^k}. \tag{24}$$

where

$$c_s^{B^k} = \sum_{i}^{\nu^k} \left((\underline{\Phi}_s^{k,i})^T \ \underline{\mathcal{I}}_3^{B^k,i} + (\underline{\Phi}_s^{k,i})^T \ \hat{\underline{\mathcal{F}}}_{G_i}^{B^k,i} \right) q_i^{B^k} + \sum_{j \in Dist[k]} \left(\left[\sum_{i}^{\nu^k} (\underline{\Phi}_s^{k,i})^T \ \hat{\underline{\Upsilon}}_i^{B_j^k,i} \right] q_i^{B^k} \cdot \underline{\mathcal{I}}_3^{J_1^j} \right), \tag{25}$$

and

$$\underline{g}^{B^k} = [\underline{g}_1^{B^k}, \ldots, \underline{g}_{\mu^{B^k}}^{B^k}]^T \triangleq (\underline{\mathcal{M}}^{B^k})^{-1} [\underline{f}_1^{B^k}, \ldots, \underline{f}_{\mu^{B^k}}^{B^k}] \tag{26}$$

with

$$\underline{f}_s^{B^k} \triangleq \left[\sum_{i}^{\nu^k} \left((\underline{\Phi}^{k,i})^T \ \underline{\mathcal{I}}_3^{B^k,i} \ (\underline{\mathcal{S}}^{B^k,i})^T + (\underline{\Phi}_s^{k,i})^T \ \hat{\underline{\Upsilon}}_i^{B^k,i} \right) q_i^{B^k} + \sum_{j \in Dist[k]} \left(\left[\sum_{i}^{\nu^k} (\underline{\Phi}_s^{k,i})^T \cdot \hat{\underline{\Upsilon}}_i^{B_j^k,i} \right] q_i^{B^k} \cdot \underline{\mathcal{I}}_3^{B^k,J_1^j} \ (\underline{\mathcal{S}}^{B^k,j})^T \right) \right] \tag{27}$$

For the rigid body degrees of freedom the expressions for $\underline{\mathcal{I}}_3^{J_j^k}$ and $\underline{\mathcal{F}}_3^{J_j^k}$ are

$$\underline{\mathcal{I}}_3^{J_j^k - 1} \triangleq \underline{\mathcal{I}}_1^{J_j^k - 1} + \underline{\mathcal{S}}^{J_j^k} \ \underline{\mathcal{I}}_3^{J_j^k} \ (\underline{\mathcal{S}}^{J_j^k})^T - \left(\frac{1}{(\underline{\mathcal{P}}^{J_j^k})^T \cdot \underline{\mathcal{I}}_2^{J_j^k} \cdot \underline{\mathcal{P}}_{J_j^k}^{J_j^k}} \right) \left[\underline{\mathcal{S}}^{J_j^k} \ \underline{\mathcal{I}}_3^{J_j^k} \ \underline{\mathcal{P}}_{J_j^k}^{J_j^k} \ (\underline{\mathcal{S}}^{J_j^k} \ \underline{\mathcal{I}}_3^{J_j^k} \ \underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T \right] \tag{28}$$

$$\underline{\mathcal{F}}_3^{J_j^k - 1} \triangleq \underline{\mathcal{I}}_1^{J_j^k - 1} + \underline{\mathcal{S}}^{J_j^k} \cdot \underline{\mathcal{F}}_3^{J_j^k} - \left(\frac{1}{(\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T \ \underline{\mathcal{I}}_2^{J_j^k} \ \underline{\mathcal{P}}_{J_j^k}^{J_j^k}} \right) \left[\underline{\mathcal{S}}^{J_j^k} \ \underline{\mathcal{I}}_3^{J_j^k} \ \underline{\mathcal{P}}_{J_j^k}^{J_j^k} \cdot (\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T \ \underline{\mathcal{F}}_3^{J_j^k} \right] \tag{29}$$

With $\underline{\mathcal{F}}_3^{B^k,J^k}$, $\underline{\mathcal{F}}_3^{J_j^k - 1}$, $\underline{\mathcal{I}}_3^{B^k,J^k}$, and $\underline{\mathcal{I}}_3^{J_j^k - 1}$ so defined, the equations of motion can be written in the form

TABLE 1, continued

Mode	Frequency	30 Hz	90 Hz	130 Hz
23	83 9	70 0	0 60	-0 10
24	84.0	206	0.75	0 04
25	85.6	225	1.38	0 04
26	85.7	232	2 83	-0.02
27	98.1	375	7.21	0.02
28	99.0	391	9 57	0.25
29	99.7	537	9.63	0 19
30	104.0	578	8.10	0.03
31	104.1	670	12.0	0 24
32	104.3	943	12.2	0.31
33	105.9	1018	10 6	0.07
34	117.3	964	19 4	0.22
35	117.5	1000	19.4	0.29
36	117.7	1043	52.3	0.38
37	120.1	1072	49.7	0.43
38	125.5	1334	44 6	0 10
39	127.4	1388	78.5	0.97
40	133 6	1453	92.6	0.00
# dof	**3360**	**46**	**62**	**78**

Thus, using CMS procedures, it is possible to significantly reduce the number of degrees of freedom as well as subdivide the global structure into substructures that may allow future modifications to be accomplished more efficiently. For example, if the required modification only involves specific components, then it is not necessary to recompute the characteristics of the unmodified components. This can lead to substantial time savings. In the telescope focus unit analysis, the effect of changing the stiffness of the bearings between the inner and outer tubes of the focus unit was investigated. Modification of this stiffness and re-analysis of the complete problem took 22% of the time required for the baseline CMS analysis. Thus in situations where many such re-analyses have to be computed in order to obtain a system with the desired response characteristics, the benefits of using a CMS procedure may be significant.

$$(\underline{\mathcal{P}}_{J_i^k}^{J_i^k})^T \left[\underline{\mathcal{I}}_3^{J_i^k}\ \bar{\mathcal{A}}^{J_i^k} - \underline{\mathcal{F}}_3^{J_i^k}\right] = 0 \quad \text{and} \quad \sum_{i}^{\mu^k} (\underline{\Phi}_s^{B^k,i})^T \left[\underline{\mathcal{I}}_3^{B^k,i}\ \bar{\mathcal{A}}^{B^k,i} + \underline{\mathcal{F}}_3^{B^k,i}\right] = 0 \quad (30)$$

fo the case where r is associated with the generalized speed $u_j^{J^k}$ and $u_s^{B^k}$ respectively

Eqns (21)-(23), (25), (27), and (30) contain summations of quantities associated with all grid points used in the finite element discretization of B^k These spatial summations (integrations) are only performed once, producing time invariant coefficients to the temporal quantities

What is important in equations of motion (30) is that the state derivatives which are in evidence in the $\bar{\mathcal{A}}$ terms are limited to those associated with the generalized coordinates of the *ancestral bodies*, the bodies which form the path between the inertial frame and the body under consideration [6, 8] Thus, working recursively inward to the lowest rigid body degree-of-freedom of the *base body*, we have

$$(\underline{\mathcal{P}}_{J_1^1}^{J_1^1})^T \left[\underline{\mathcal{I}}_3^{J_1^1}\ \bar{\mathcal{A}}^{J_1^1} + \underline{\mathcal{F}}_3^{J_1^1}\right] = 0 \qquad \text{where} \qquad \bar{\mathcal{A}}^{J_1^1} = \underline{\mathcal{P}}_{J_1^1}^{J_1^1} u_1^{J_1^1} \qquad (31)$$

Substituting the second expression (31) into the first yields

$$\dot{u}_1^{J_1^1} = -\frac{(\underline{\mathcal{P}}_{J_1^1}^{J_1^1})^T \cdot \underline{\mathcal{F}}_3^{J_1^1}}{(\underline{\mathcal{P}}_{J_1^1}^{J_1^1})^T \cdot \underline{\mathcal{I}}_2^{J_1^1} \cdot \underline{\mathcal{P}}_{J_1^1}^{J_1^1}} \qquad (32)$$

providing the value of $u_1^{J_1^1}$ in terms of known quantities.

Recursive Back-substitution. This known value for $u_1^{J_1^1}$ is used to start the back-substitution process for the determination of the remaining generalized speed derivatives. The recursive relationships used here are

$$\bar{\mathcal{A}}^{J_j^k} = (\underline{\mathcal{S}}^{J_j^k})^T \cdot \bar{\mathcal{A}}^{J_{j-1}^k} + \underline{\mathcal{P}}_{J_j^k}^{J_j^k} u_j^{J_j^k} \qquad (33)$$

and

$$u_j^{J_j^k} = \frac{-(\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T}{(\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T\ \underline{\mathcal{I}}_2^{J_j^k}\ \underline{\mathcal{P}}_{J_j^k}^{J_j^k}} \left[\underline{\mathcal{I}}_3^{J_j^k}\ (\underline{\mathcal{S}}^{J_j^k})^T\ \bar{\mathcal{A}}^{J_{j-1}^k} + \underline{\mathcal{F}}_3^{J_j^k}\right] \qquad (34)$$

for rigid body degrees of freedom and

$$\bar{\mathcal{A}}^{B^k,i} = (\underline{\mathcal{S}}^{B^k,i})^T\ \bar{\mathcal{A}}^{B^k,J^k} + \underline{\Phi}_s^{k,i} \dot{u}_s^{B^k} \qquad \text{with} \qquad u_s^{B^k} = d_s^{B^k} + \underline{g}_s^{B^k}\ \bar{\mathcal{A}}^{B^k,J^k} \qquad (35)$$

for flexible body degrees of freedom $u_1^{J^1}$ is substituted into Eq.(33) for the determination of $\bar{\mathcal{A}}^{J_1^1}$, which is in turn used in Eq (34) for the determination of $\dot{u}_2^{J^1}$ This

process is repeated working recursively outward to determine the remaining joint accelerations, with the state derivatives associated with the flexible degrees of freedom determined through the use of Eqs.(35) The end result is that the rigid body degrees of freedom are uncoupled and the relative joint accelerations, $u_j^{J^*}$ ($k = 1$ ..N and $j = 1, \ldots, \mu^{J^*}$) are determined in $O(N)$ operations overall for a general tree structure

2.4 EXTENSION TO CONSTRAINED SYSTEMS

The dynamical equations presented so far apply only to tree configurations which are not subject to either motion or configuration constraints If a system has one or more constraints on its geometry or motion, then additional equations must be satisfied.

Consider an unconstrained multibody dynamical system possessing n degrees of freedom. The motion of the system is fully specified by the generalized speeds, $u_1, \ldots, u_n$, which are independent of each other If the system is subjected to m conditions of the form

$$\Phi_i(q,t) = 0 \qquad \text{and} \qquad \Phi_j(q,q,t) = 0 \qquad (i + j = 1, \ldots, m) \tag{36}$$

for holonomic and nonholonomic constraints, respectively, then the motion of the system in a Newtonian reference frame is characterized by n generalized speeds $u_1, \ldots, u_n$ which are not independent of each other, but must satisfy m simple nonholonomic constraints, or holonomic constraints differentiated once with respect to time. Such constraint equations are of the form

$$\sum_{r=1}^{n} A_{rs} u_r + B_s = 0 \qquad (s = 1, \ldots, m) \tag{37}$$

where A_{rs} and B_s are explicit functions of $q_1, \ldots, q_n$ and time t As a consequence of the imposition of these constraints, the number of degrees of freedom of the system reduces from n to $p \triangleq n - m$.

The procedure presented here for the formulation of the equations of motion for systems subject to constraints is a variation on that presented by Park [15,16], and applied to an $O(n)$ approach in Refs [6] and [8].

It can be shown [8] that the equations of motion for the constrained system can be written in the form

$$\underline{\dot{u}} + \underline{\Gamma}\,\underline{f_c} = \underline{\eta} \qquad \text{subject to} \qquad \underline{A}\,\underline{u} + \underline{B} = \underline{0} \tag{38}$$

Solving the first expression in (38) for $\underline{u}$ and substituting this into the first time derivative of the penalty equation,

$$\underline{f_c} = \frac{1}{\epsilon}(\underline{A}\,\underline{\dot{u}} + \underline{\dot{A}}\,\underline{u} + \underline{\dot{B}}), \tag{39}$$

yields

$$\epsilon \underline{f_c} + \underline{A}\,\underline{\Gamma}(\underline{f_c} + k_1\underline{\Phi} + k_2\underline{\dot{\Phi}}) = \underline{A}\,\underline{\eta} + \underline{\dot{A}}\,\underline{u} + \underline{\dot{B}} \tag{40}$$

where ϵ is a positive constant chosen by the analyst, with k_1 and k_2 being control gains for stabilization of constraint violation error.

The solution of this ordinary differential equation in $\underline{f}_c$ decays to the constraint load measure number values When constraints are present, the procedure follows much the same course as the basic algorithm presented in the previous section, but now the presence of the unknown constraint force measure numbers in the equations of motion must be considered This algorithm, like most others, requires that the system be a tree structure. This necessitates that closed loops be cut at the joints connecting appropriate bodies so that the required open loop structure is produced. The constraint conditions which insure closure of the loops are then enforced through the addition of constraint forces and moments of proper magnitude and direction, applied at the connection points. In the basic algorithm, the set of all distance and contact forces acting on $P^{k,s}$ are given by the reaction $\mathbf{R}^{B^k,s}$ and torque $\mathbf{T}^{B^k,s}$ The presence of the constraint forces $\mathbf{R}_c^{B^k,h}$ and constraint torques $\mathbf{T}_c^{B^k,h}$ must now also be considered.

From these we define

$$\underline{\mathcal{F}}_{c_1}^{B^k,h} \triangleq \begin{bmatrix} \mathbf{T}_c^{B^k,h} \\ \mathbf{R}_c^{B^k,h} \end{bmatrix} \tag{41}$$

where h is the point of B^k through which a constraint is applied.

Defining the quantities $d_c^{B^k}$, $\underline{\mathcal{F}}_{c_3}^{B^k,h}$ and $\underline{\mathcal{F}}_{c_3}^{J_{j-1}^k}$ as

$$d_{c_s}^{B^k} \triangleq (\underline{\mathcal{M}}^{B^k})^{-1}\left(\sum_h\left(\left[\sum_t^{\nu^k}(\underline{\Phi}_s^{k,t})^T\ \underline{\Upsilon}_t^{B_t^k,t}\right]q_t^{B^k}\ \underline{\mathcal{F}}_{c_1}^{B^k,h}\right)\right. \\ \left. + \sum_{j\in Dist[k]}\left(\left[\sum_t^{\nu^k}(\underline{\Phi}_s^{k,t})^T\ \underline{\Upsilon}_t^{B_j^k,t}\right]q_t^{B^k}\ \underline{\mathcal{F}}_{c_3}^{J_1^j}\right)\right), \tag{42}$$

$$\underline{\mathcal{F}}_{c_3}^{B^k,h} \triangleq \sum_h\left(\underline{\mathcal{S}}^{B^k,h}\cdot\underline{\mathcal{F}}_{c_1}^{B^k,h} + \underline{\mathcal{S}}^{B^k,h}\cdot\underline{\mathcal{I}}_3^{B^k,h}\ \underline{\Phi}_s^{k,h}d_{c_s}^{B^k}\right. \\ \left. + \underline{\mathcal{S}}^{B^k,h}\cdot\underline{\Upsilon}_t^{B_j^k,h}q_t^{B^k}\ \underline{\mathcal{F}}_{c_1}^{B^k,h}\right) + \sum_{j\in Dist[k]}\left(\underline{\mathcal{S}}^{B^k,1}\ \underline{\Upsilon}_t^{B_j^k,1}q_t^{B^k}\ \underline{\mathcal{F}}_{c_3}^{B^k,J_0^j}\right) \tag{43}$$

$$\underline{\mathcal{F}}_{c_3}^{J_{j-1}^k} \triangleq \underline{\mathcal{F}}_{c_1}^{J_{j-1}^k} + \underline{\mathcal{S}}^{J_j^k}\ \underline{\mathcal{F}}_{c_3}^{J_j^k} - \left(\frac{1}{(\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T\ \underline{\mathcal{I}}_2^{J_j^k}\ \underline{\mathcal{P}}_{J_j^k}^{J_j^k}}\right)\left[\underline{\mathcal{S}}^{J_j^k}\ \underline{\mathcal{I}}_3^{J_j^k}\ \underline{\mathcal{V}}_{J_j^k}^{J_j^k}\ (\underline{\mathcal{P}}_{J_j^k}^{J_j^k})^T\ \underline{\mathcal{F}}_{c_3}^{J_j^k}\right], \tag{44}$$

the state derivatives for the constrained system are given by

$$u_j^{J^k} = \frac{-(\underline{\mathcal{P}}_j^{J^k})^T}{(\underline{\mathcal{P}}_{J_j^k}^{J^k})^T \ \underline{\mathcal{I}}_2^{J^k} \ \underline{\mathcal{P}}_{J_j^k}^{J^k}} \cdot \left[\underline{\mathcal{I}}_3^{J^k} \ (\underline{\mathcal{S}}^{J^k})^T \ \underline{\bar{\mathcal{A}}}^{J^k_{J-1}} + \underline{\mathcal{F}}_3^{J^k} + \underline{\mathcal{F}}_{c3}^{J^k}\right] \tag{45}$$

and

$$\underline{u}^{B^k} = \underline{d}^{B^k} + \underline{d}_{c_3}^{B^k} + \underline{g}^{B^k} \ \underline{\bar{\mathcal{A}}}^{B^k,J^k} \tag{46}$$

Expressing $\underline{u}$ as

$$\dot{\underline{u}} = \underline{\eta} + \underline{\zeta} \tag{47}$$

where $\underline{\zeta}$ is that portion of $\underline{u}$ which is explicit in the constraint load measure numbers and $\underline{\eta}$ is all else; $\underline{\eta}$ is simply equal to $\dot{\underline{u}}$ for the identical system when no constraints are imposed and is given by eqs (33)-(35). In a similar manner, the elements of $\underline{\zeta}$ are given by the relations

$$\zeta_j^{J^k} = \frac{-(\underline{\mathcal{P}}_{J_j^k}^{J^k})^T}{(\underline{\mathcal{P}}_{J_j^k}^{J^k})^T \cdot \underline{\mathcal{I}}_2^{J^k} \cdot \underline{\mathcal{P}}_{J_j^k}^{J^k}} \cdot \left[\underline{\mathcal{I}}_3^{J^k} \ (\underline{\mathcal{S}}^{J^k})^T \cdot \underline{\bar{\zeta}}^{J^k_j} + \underline{\mathcal{F}}_3^{J^k_j} + \underline{\mathcal{F}}_{c3}^{J^k_j}\right] = -\sum_{i=1}^{m} \Gamma_i^{J^k_j} f_c^{J^k_j}, \tag{48}$$

where

$$\underline{\bar{\zeta}}^{J^k_j} = (\underline{\mathcal{S}}^{J^k_j})^T \cdot \underline{\bar{\zeta}}^{J^k_{j-1}} + \underline{\mathcal{P}}_{J_j^k}^{J^k_j} \zeta_j^{J^k} \quad \text{and} \quad \zeta_j^{B^k} = \underline{d}_{c_j}^{B^k} + \underline{g}_j^{B^k} \ \underline{\bar{\zeta}}^{B^k,J^k} = -\sum_{i=1}^{m} \Gamma_{B^k,i} f_{c_i} \tag{49}$$

with

$$\underline{\bar{\zeta}}^{B^k,j} \triangleq (\underline{\mathcal{S}}^{B^k,j})^T \ \underline{\bar{\zeta}}^{B^k,J^k} + \underline{\Phi}_s^{k,j} \zeta_s^{B^k} \tag{50}$$

So, the equations of motion for the constrained system may be rewritten as

$$\dot{\underline{u}} + \underline{\Gamma} \underline{f}_c = \underline{\eta} \tag{51}$$

where $\underline{\Gamma}$ is defined by (48) and (49), and $\underline{f}_c$ is obtained from Eq.(40). The state derivatives are then calculated from Eq (51), with the total procedure requiring approximately $O(n + m^3)$ operations for rigid body systems.

3. Numerical Examples

Two example are provided to validate the formulation and offer evidence of the improved performance possible when using an $O(n)$ formulation for the analysis of systems containing a large number of bodies.

3.1 CANTILEVER BEAM WITH PRESCRIBED BASE MOTION

The system discussed here consists of a cantilever beam attached to a base with prescribed angular velocity - an example used as a benchmark test by a number of authors. Refs. [14,17-19]. The generalized coordinates used in this problem are the four modal coordinates and the angular velocity of the base is given by

$$\omega(t) = \begin{cases} \frac{2}{5}\left[t - \frac{7.5}{\pi}\sin\left(\frac{\pi t}{7.5}\right)\right] & \text{if } t < 15 \text{ sec} \\ 6.0 \text{ (rad/sec)} & \text{if } t \geq 15 \text{ sec} \end{cases} \tag{52}$$

The beam geometry and properties which appear in this example were those used by Ryan [18], as well as Wallrapp and Schwertassek [19]

Figure 2 shows the tip deflection of the beam central axis from its undeformed position during the course of a 20 second simulation. The results agree well with those obtained using the formulation presented in Ref. [17].

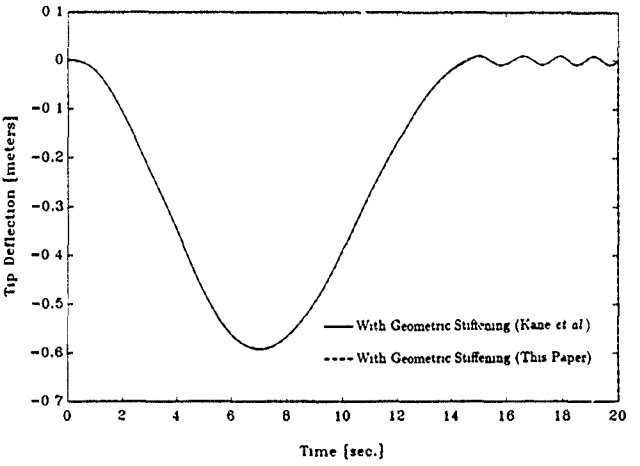


Fig. 2: Tip deflection of rotating cantilever beam

3.2 $\frac{n}{2}$-BODY CHAIN WITH CONSTRAINED ENDS

The system consists of a chain of rigid bodies with each end of the chain connected to a point fixed in an inertial reference frame. The $\frac{n}{2}$ bodies of the chain are connected

to each other and to points fixed in the inertial frame by two degree of freedom Hooke's joints.

If n represents the number of degrees of freedom associated with the unconstrained system, then the system has $\frac{n}{2}$ bodies, $\frac{n}{2}+1$ joints, and the actual number of system degrees of freedom is $p = n - 3$. The motion of the system was simulated for $n = 4$, 12, 20, 28, 36 and 44.

Results obtained from simulation codes produced using the $O(n)$ algorithm presented in this paper were verified through direct comparison with results obtained for identical systems using simulation codes written from equations of motion derived using a more traditional $O(n^3)$ procedure.

The simulation results obtained by each of the two formulations were effectively identical. However, as indicated in Fig. 3, the CPU times required in performing a desired 1.0 second simulation differ markedly for each of these formulations when n is large. It is readily seen from the figure that substantial saving in computer time and associated cost are possible when using an $O(n)$ formulation relative to more conventional $O(n)^3$ formulations.

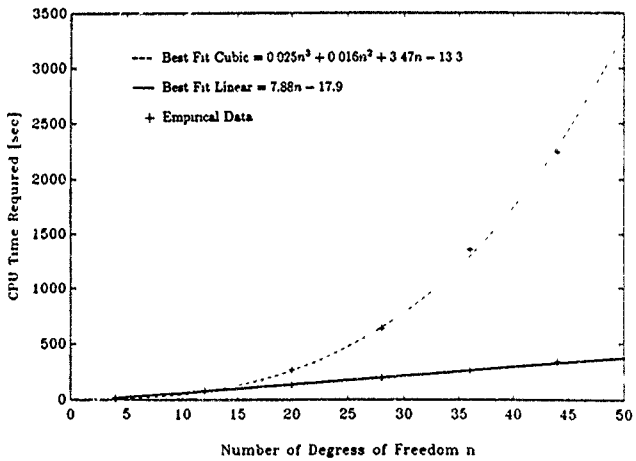


Fig 3: CPU time requirement as a function of n

4.0 Concluding Remarks

A general formulation is presented for the analysis of transient response of multibody systems with flexible members The method uses admissible shape functions derived from finite element modeling of the component members and thus allows the modeling of flexibility for general bodies The formulations treats the general case of coupled large rigid body displacements, linear elastic deformation, and includes a first order representation geometric stiffening effects.

The formulation presented applies to systems involving three-dimensional motions, which may be comprised of any joint type which can be modeled as a series of revolute and prismatic joints. The equations of motion produced in this manner are uncoupled in the rigid body degrees of freedom, with coupling only existing between the flexible degrees of freedom associated with the individual flexible bodies. Furthermore, the equations are generated in a form which exploits the coarse grain concurrency of the mathematical model to the maximum degree. Thus, the computer simulation code produced from this type of formulation is particularly well suited for application on some forms of parallel computers.

REFERENCES

[1] Hooker, W. W. and Margulies, G., "The Dynamical Attitude Equations for an n-Body Satellite," *Journal of the Astronomical Sciences*, Vol 7, No 4, 1965, pp. 123-128, .

[2] Walker, M. W , and Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *Journal of Dynamic Systems, Measurement , and Control*, Vol. 104, 1982, pp. 205-211.

[3] Luh, J. S. Y., Walker, M. W. and Paul, R P. C., "On-Line Computational Scheme for Mechanical Manipulators," *Journal of Dynamic Systems, Measurements, and Control*, Vol. 102, 1980, pp. 69-76

[4] Kim, S. S., and Haug, E. J., "Recursive Formulation for Flexible Multibody Dynamics· Part I, Open-Loop Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 71, No 3, 1988, pp. 293-314.

[5] Kim, S. S., and Haug, E. J., "Recursive Formulation for Flexible Multibody Dynamics· Part II, Closed-Loop Systems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 74, No. 3, 1989, pp. 251-269.

[6] Anderson, K. S , "Recursive Derivation of Explicit Equations of Motion for Efficient Dynamics/Control Simulation of Large Multibody Systems," Ph.D. Dissertation, Stanford University, July, 1990.

[7] Rosenthal, D E. . An Order n Formulation for Robotic Systems. *The Journal of the Astronautical Sciences* Vol. 38, No 4 1990, pp 511-529

[8] Anderson, K. S . "An Efficient Formulation for the Modeling of General Multi-Flexible-Body Constrained Systems." *International Journal of Solids and Structures*, Vol 30, No 7, pp. 921-945, 1993.

[9] Banerjee, A. K. "Multibody Dynamics of Systems with Flexible Components. 2nd European Space Agency (ESA) Workshop, *Modal Representation of Flexible Structures By Continuum Methods*. ESTEC, Noordwijk, The Netherlands, June 3-5, 1992.

[10] Kane, T. R., and Levinson, D A *Dynamics: Theory and Applications*, McGraw-Hill Book Co., New York, 1985.

[11] Craig, R. R., and Bampton, M. C. C., "Coupling of Substructures for Dynamic Analysis" *AIAA journal*, Vol 6, 1968, pp. 1313-1319.

[12] Bakr, E. M., and Shabana, A A., "Geometrically Nonlinear Analysis of Multi-body Systems," *Computers and Structures*, Vol. 23, No 6, 1986, pp. 739-751

[13] Ider, S. K., and Amirouche, F. M. L., "The Influence of Geometric Nonlinearities in the Dynamics of Flexible Tree-Like Structures," *Journal of Guidance, Control, and Dynamics*, Vol. 12, 1989, pp. 830-837.

[14] Banerjee, A. K., and Dickens, J. M., "Dynamics of an Arbitrary Flexible Body in Large Rotation and Translation," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 2, 1990, pp. 221-227

[15] Park, K. C., and Chiou, J. C., "Stabilization of Computational Procedures for Constrained Dynamical Systems," *Journal of Guidance*, Vol. 11, No 4, pp 365-370, 1988

[16] Park, K. C. and Chiou, J. C., and Downer, J. D., "An Explicit-Implicit Staggered Procedure for Multibody Dynamic Analysis," *Journal of Guidance, Control and Dynamics*, Vol. 13, No. 3, May-June 1989, pp. 562-570.

[17] Kane, T. R., Ryan, R. R., and Banerjee, A. K. , "Dynamics of a Cantilever Beam Attached to a Moving Base," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 2, 1987, pp 139-151.

[18] Ryan, R. R , "Flexible Multibody Dynamics· Problems and Solutions," *SDIO/NASA Workshop on Multibody Dynamics*, Jet Propulsion Laboratory, Pasadena, CA , 1987

[19] Wallrapp, O. and Schwertassek, R., "Geometric Stiffening in Multibody Systems Simulation," *Proceedings of Int Forum on Aeroelasticity and Structural Dynamics 1991*, Aachen, DGLR, D-3300 Bonn

[20] Glück, R., "A Custom Architecture Parallel Processing System for the Space Station," Report TRW No. EML-003, TRW, Redondo Beach, CA, 1989

[21] Hwang, R. S., Bae, D S., Kuhl, J. G and Haug, E. J., "Parallel Processing for Real-Time Dynamic System Simulation," Center for Simulation and Design Optimization, University of Iowa, 1988.

AN ENERGY CONSERVING CO-ROTATIONAL PROCEDURE FOR NON-LINEAR DYNAMICS WITH FINITE ELEMENTS

M A CRISFIELD[1] & J. SHI[2].

1 Dept. of Aeronautics, Imperial College, London
2 Dept. of Aeronautics, Queen's University, Belfast

ABSTRACT

A new procedure is proposed for implicit dynamic analysis using the finite element method. The main aim is to give stable solutions with large time-steps in the presence of significant rigid body motions, in particular rotations. In contrast to most conventional approaches, the time integration strategy is closely linked to the "element technology" with the latter involving a form of co-rotational procedure. For the undamped situation, the solution procedure leads to an algorithm that exactly conserves energy when constant external forces are applied (i.e. with gravity loading).

1. Introduction

Stewart [1] has highlighted the potential dangers of false solution from time-integration procedures with non-linear dynamics. For the finite element method, conventional procedures can be divided into two main types: explicit and implicit. The former is traditionally used for wave-propagation and high-velocity impact problems while the latter is more appropriate for structural applications dominated by the low frequency response although Dow.. et al [2] have recently described an energy conserving explicit algorithm that gives excellent results for the latter problems. The present paper is devoted to implicit methods.

Conventional techniques adopt very similar predictor/corrector procedures to those used for non-linear statics. However, the out-of-balance force vector is now augmented by the mass-times-acceleration terms while the conventional static tangent stiffness matrix is augmented by a factor times the mass matrix if, as here, no damping is considered. Dynamic equilibrium is usually enforced at the end of the step.

Various time-integration strategies can be used for up-dating the velocities and accelerations with the Newmark methods [3-5] often being used. In the present paper, the average acceleration method (or trapezoidal rule with $\beta = 1/4$ and $\gamma = 1/2$) will be used to represent the "conventional procedure". The latter technique is "unconditionally stable" (irrespective of the time step) for linear problems. However, this stability does not extend to non-linear systems and consequently, as will be shown in the paper, the 'conventional method" often requires absurdly small time steps if the solution is not to "lock" onto a higher

energy state (see later) Attempts to overcome these difficulties have been made by both Haug et al [7] and Hughes et al [8] who used Lagrangian multipliers to enforce an energy constraint

In a standard algorithm, the time-integration procedure is totally divorced from the "element technology" (with the exception of the provision of the mass matrix) Most implicit dynamic procedures satisfy the dynamic equilibrium at the end of the step. More recently, Simo and co-workers have explored the idea of a "mid-point equilibrium" [9,10] which follows on from the work of Hilber et al [11] and Zienkiewicz et al [12] These concepts will be taken further in the present paper.

A key feature relates to the precise definition of the stresses and strains that are associated with the "mid-point" configuration. The present method relies heavily on co-rotational concepts [6,13]. The latter have usually been used for statics with the key aim being to divorce the straining from the rigid body motions, in particular the rigid-body rotation. The authors have extended the ideas to dynamics [14,15] and, to this end, have used both a mid-point configuration and "averaged strains" Similar ideas (although without the co-rotational framework) have recently been described by Simo and Tarnow [16,17] in order to develop algorithms that conserve energy when constant external forces are applied.

In the first part of the paper, the theory is developed for simple two-dimensional truss elements and this is followed by a set of examples involving such elements. In the final part of the paper, the theory is described for the extension to two-dimensional beams and, in addition, ideas are outlined for the extension to three-dimensional beams and shells.

2. Motivation

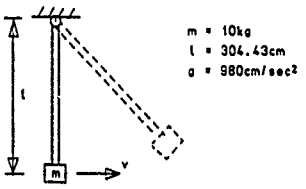


Fig. 1 Bathe's pendulum

The work was triggered by the authors first attempts to apply dynamics to the simple pendulum shown in Fig. 1 (with EA = 10^{10}N). This problem has been discussed in detail by Bathe in his book [4] and study guide [18]. In both cases, the pendulum was dropped from the horizontal position, while in the present analysis, it is "fired" with an initial horizontal velocity of 772.5 cm/sec. from the vertical position The basic behaviour is very similar.

From the response in Fig. 2, one can see that although the period is of the order of 4.0 seconds "conventional procedures" (as previously defined) require a time step of the order of 0.025 seconds to give a satisfactory solution.

Figure 2 shows that if a time step of 0.1 seconds is adopted, the solution "locks" at a position close to the initial position after one and a quarter periods Fig 3 shows that, prior to this 'locking', there has been a build-up of energy

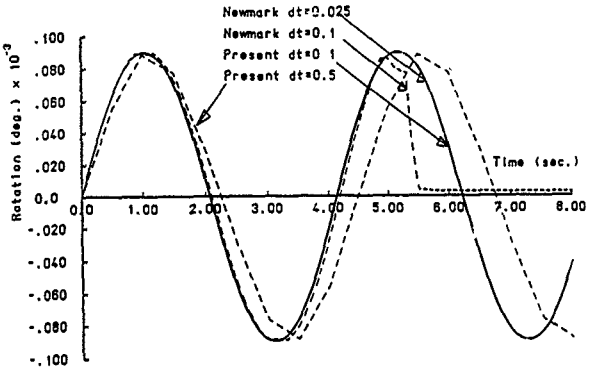


Fig. 2 Relationships between rotation and time for Bathe's pendulum

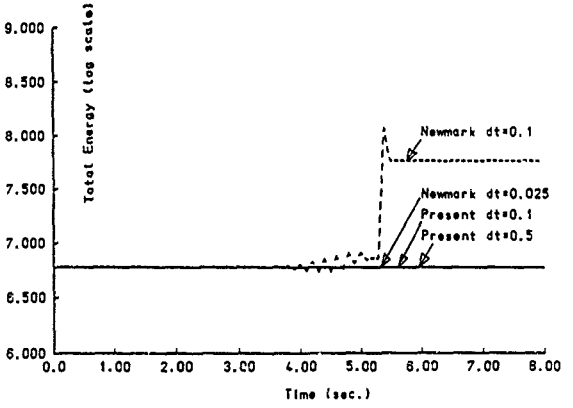


Fig 3 Relationships between energy and time for Bathe's pendulum

with an increasing percentage going into strain energy associated with an oscillating axial strain (Fig. 4) This phenomenon was earlier noted by Bathe [18] The term 'locking is used because, although the bar still moves, the overall (lower mode) motions are now fairly small while the local axial stretching (higher mode) dominates

It would be possible to overcome these difficulties for this simple pendulum by switching variables to use the rotation and stretch.. However, the procedure could not then be easily extended to other configurations within a conventional finite element framework. Before detailing the theory for the new method, we note that, for this problem, it leads to a very satisfactory solution with a time step of 0.1 (Fig. 2 and 3) and that even with a time step of 0.5, no locking is encountered (Figs 2 and 3)

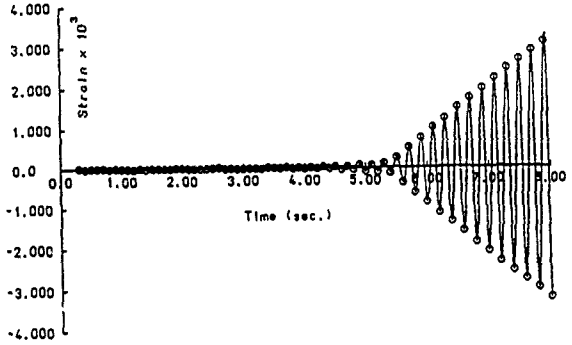


Fig. 4 Relationships between axial strain and time for Bathe's pendulum

3. Theory

Before describing the new method, a brief description is given of the "conventional" trapezoidal rule (or average acceleration method) The displacements, d, are chosen as the "driving variables" so that the velocity at the end of the step, v_{n+1}, can be obtained via.

$$v_{n+1} = \frac{2}{\Delta t}(d_{n+1} - d_n) - v_n = \frac{2}{\Delta t}\Delta d - v_n \qquad (1)$$

while the acceleration is then obtained from

$$a_{n+1} = \frac{2}{\Delta t}(v_{n+1} - v_n) - a_n = \frac{2}{\Delta t}\Delta v - a_n \qquad (2)$$

Without damping, the equilibrium equations at the end of the step are given by

$$g_{n+1} = q_{i,n+1} + Ma_{n+1} - q_{e,n+1} = 0 \qquad (3)$$

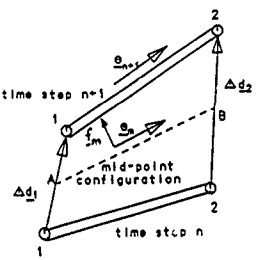


Fig. 5 Two-noded truss element

where q_i are the (static) internal forces, M is the mass matrix and q_e are the external forces. Equations (3) may be solved using the Newton–Raphson method in which the linearisation of (3) is used to obtain the effective tangent stiffness matrix.

In the following, we will concentrate on the equations for a simple two noded truss element as shown in Fig. 5. The external forces and the masses will be assumed fixed and related to the element thereby allowing us to concentrate on the latter. In these circumstances, with the co-rotational approach [6], equations (3) become

$$g_{n+1} = N_{n+1}\begin{bmatrix} -e_{n+1} \\ e_{n+1} \end{bmatrix} + \frac{1}{\Delta t}\begin{bmatrix} m_1\Delta v_1 \\ m_2\Delta v_2 \end{bmatrix} - \begin{bmatrix} m_1 a_1 \\ m_2 a_2 \end{bmatrix} - \begin{bmatrix} q_{e1} \\ q_{e2} \end{bmatrix} = 0 \qquad (4)$$

where N_{n+1} is the axial force (tensile positive) in the bar at the end of the step and e_{n+1} is the unit vector lying along the bar (Fig. 5).

In Ref. (9), Simo and co-workers have proposed proposed a "mid-point procedure" in which, for the up-dating, equation (1) is retained but equation (2), which can be re-expressed as:

$$a_{av} = \frac{1}{\Delta t}\Delta v \qquad (5)$$

is replaced by

$$a_m = \frac{1}{\Delta t}\Delta v \qquad (6)$$

with the end-point equilibrium of (3) being replaced by the mid-point equilibrium of:

$$g_m = q_{i\,m} + Ma_m - q_{e,m} = 0 \qquad (7)$$

With the previous assumptions, for the bar element of Fig. 5, this leads, in place of (4) to

$$g_m = N_m \begin{Bmatrix} -e_m \\ e_m \end{Bmatrix} + \frac{1}{\Delta t}\begin{Bmatrix} m_1 \Delta v_1 \\ m_2 \Delta v_2 \end{Bmatrix} - \begin{Bmatrix} q_{e1} \\ q_{e2} \end{Bmatrix} = 0 \qquad (8)$$

The key factor involves the definition of N_m via its associated strain. In Ref (9), Simo and co-workers obtained the mid-point strain directly from the mid-point configuration. However, as illustrated in Fig. 6, for the previous pendulum problem, this leads to high artificial straining.

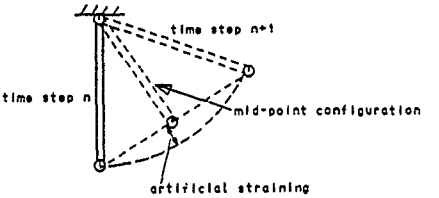


Fig 6 Artificial straining with conventional mid-point procedure

In order to remove the artificial straining, in [14,15], the authors simply redefined N_m as a factor β (which will be defined later) times the average of the axial forces at the beginning and ends of the step, i.e.:

$$N_m = \beta\,(N_n + N_{n+1})/2 = \beta N_{av} \qquad (9)$$

Simo and Tarnow [16,17] have adopted a similar approach (although without the factor β for reasons that will be discussed later).

To complete the definition of the the terms in (8), the force in the bar at the current configuration is obtained via:

$$N_{n+1} = N_{init} + \frac{EA}{l_o}(l_{n+1} - l_o) \qquad (10)$$

with l_o as the initial length of the bar. The current length is simply obtained using:

$$l_{n+1}^2 = x_{21,n+1}^T x_{21,n+1} = (X_{21} + d_{21})^T (X_{21} + d_{21}) \qquad (11)$$

where, for example $d_{21} = d_2 - d_1$ and X_1 and X_2 are the initial position vectors of nodes 1 and 2. In addition, the mid-point unit vector, e_m in (8) is obtained as

$$e_m = (x_{21,n} + \tfrac{1}{2}\Delta d_{21})/l_{mt} \qquad (12)$$

where the true mid-point length, ℓ_{mt} (between points A and B in Fig 5) is given by

$$\ell_{mt} = \|x_{21,n} + \tfrac{1}{2}\Delta d_{21}\| \qquad (13)$$

(The procedure breaks down if the incremental rotation is 180o when ℓ_{mt} becomes zero if there is no stretching. Such enormous increments are unlikely to be used). The factor β in (9) is provided to ensure energy conservation.

Using (1) the change in kinetic energy at a node can be obtained as:

$$\Delta K = \tfrac{1}{2}m(v_{n+1}^T v_{n+1} - v_n^T v_n) = m v_{av}^T \Delta v = \tfrac{1}{\Delta t} m \Delta v^T \Delta d \qquad ..(14)$$

while with the previous assumptions, the change in external potential energy at a node is,

$$\Delta P = - q_e^T \Delta d \qquad (15)$$

For an element, the change in strain energy is given by,

$$\Delta\phi = \frac{\ell_o}{2EA}(N_{n+1}^2 - N_n^2) = \frac{\ell_o}{EA} N_{av} \Delta N \qquad (16)$$

where:

$$\Delta N = \frac{EA}{\ell_o}(\ell_{n+1} - \ell_n) = \frac{EA}{\ell_o}\frac{(\ell_{n+1}^2 - \ell_n^2)}{(\ell_{n+1} + \ell_n)} \qquad (17)$$

while

$$\ell_{n+1}^2 - \ell_n^2 = 2x_{21,n}^T \Delta d_{21} + d_{21}^T d_{21} = 2\ell_{mt} e_m^T \Delta d_{21} \qquad (18)$$

with ℓ_{n+1} having been previously defined in (13) From (16)-(18), we obtain

$$\Delta\phi = \beta N_{av} e_m^T \Delta d_{21} \qquad (19)$$

with

$$\beta = \frac{\ell_{n+1}}{\ell_{mav}} = \frac{2\ell_{n+1}}{(\ell_n + \ell_{n+1})} \qquad (20)$$

Hence, with the mid-point axial force being given by (9) with β from (20), the total energy change, can be expressed as

$$\Delta\Phi = \Delta\varphi + \Delta K + \Delta P = g^T \begin{Bmatrix} \Delta d_1 \\ \Delta d_2 \end{Bmatrix} \qquad (21)$$

with g from (8). At equilibrium g is zero and hence, at equilibrium, energy is exactly preserved. It is worth noting that the factor β tends to unity as the step-size tends to zero and is also unity when there is no rotation

Equations (7)–(9) can be linearised to give·

$$\delta g_m = \delta q_{i,m} + \frac{1}{\Delta t} \begin{Bmatrix} m_1 \delta v_1 \\ m_2 \delta v_2 \end{Bmatrix} = \bar{K}_t \delta d \qquad (22)$$

where, via (1)

$$\bar{K}_t = K_t + \frac{2}{\Delta t^2} \begin{bmatrix} m_1 I & 0 \\ 0 & m_2 I \end{bmatrix} \qquad (23)$$

and·

$$\delta q_{i,m} = \delta N_m \begin{Bmatrix} -e_m \\ e_m \end{Bmatrix} + N_m \begin{Bmatrix} -\delta e_m \\ \delta e_m \end{Bmatrix} = K_t \delta d \qquad (24)$$

Full details are given in [14]. It should be noted that K_t is non-symmetric although an artificial symmetrisation can be applied if the time steps are not too large [14].

3.1 AN ALTERNATIVE FORMULATION

Following on from the work in [16], Simo and Tarnow [17] have very recently proposed an energy preserving procedure for shells. We will now attempt to establish links with the present formulation.

If a total Lagrangian formulation is adopted, equation (4) becomes [6]

$$g_{n+1} = N_{n+1} b_{n+1} + \frac{1}{\Delta t} \begin{Bmatrix} m_1 \Delta v_1 \\ m_2 \Delta v_2 \end{Bmatrix} - \begin{Bmatrix} m_1 a_1 \\ m_2 a_2 \end{Bmatrix} - \begin{Bmatrix} q_{e1} \\ q_{e2} \end{Bmatrix} = 0 \qquad (25)$$

where

$$b_{n+1} = \frac{l_{n+1}}{l_o} \begin{Bmatrix} -e_{n+1} \\ e_{n+1} \end{Bmatrix} \qquad (26)$$

and in place of (10).

$$N_{n+1} = N_{init} + \frac{EA}{2 l_o^2} (l_{n+1}^2 - l_o^2) \qquad (27)$$

Applying the approach of Simo and Tarnow [17], in place of (8), the mid-point

equilibrium would then take the form:

$$g_m = N_{av}\,\frac{1}{2}(b_n+b_{n+1}) + \frac{1}{\Delta t}\begin{Bmatrix} m_1\Delta v_1 \\ m_2\Delta v_2 \end{Bmatrix} - \begin{Bmatrix} q_{e1} \\ q_{e2} \end{Bmatrix} = 0 \qquad (28)$$

where N_{av} is simply the mean of the N's at configurations n and n+1 (as before but with (27) now defining the N's). Following the lines adopted previously, it is easy to show that this approach leads to an energy conserving algorithm. It can also be shown that, provided the strains (but not the deformations or rotations) are small, we can adopt equation (28) for the present co-rotational formulation with the N's being given by (10) and that this leads to a procedure that is very close to that given previously. By comparing (28) with (8) and (9), it is clear that we merely need to establish that:

$$\frac{1}{2}(b_n + b_{n+1}) = \frac{1}{2}\left(\begin{bmatrix} -e_n \\ e_n \end{bmatrix} + \begin{bmatrix} -e_{n+1} \\ e_{n+1} \end{bmatrix}\right) \simeq \frac{\ell_{mt}}{\ell_{mav}}\begin{bmatrix} -e_m \\ e_m \end{bmatrix} = \beta\begin{bmatrix} -e_m \\ e_m \end{bmatrix} \qquad (29)$$

where we are now using the co-rotational form for the b vector (see (4) and [6]). With small strains, it is easy to establish that this relationship does indeed hold because

$$\frac{1}{2}(e_n + e_{n+1}) = \frac{\ell_n+\ell_{n+1}}{2\ell_n\ell_{n+1}}\left(x_{21,n} + \frac{\ell_n}{\ell_n+\ell_{n+1}}\Delta d_{21}\right)$$

$$\simeq \frac{1}{\ell_{mav}}\left(x_{21,n} + \frac{1}{2}\Delta d_{21}\right) = \frac{\ell_{mt}}{\ell_{mav}}e_m = \beta e_m \qquad (30)$$

4. Numerical Examples

The following examples have been computed using the original formulation [14,15] rather than the previous "alternative formulation". For the first set of examples which involve small strains, we would expect almost identical solutions via the "alternative formulation"

The numerical results for Bathe's pendulum have already been described. In the first set of following examples, Young's modulus times the area had been chosen to be 10^{10}N so that the bars or chain-links are nearly rigid. In all cases, fixed time increments have been employed. The terminology "Newmark" refers to the Newmark scheme with $\beta = 0.25$ and $\gamma = 0.5$.

4.1 FOUR-CHAIN SYSTEM

Fig. 7 shows the computed response for a four-bar chain of the same EA value as the previous pendulum and of total length 400 which is free to fly (in the absence of gravity). The chain was given an initial linearly varying downward velocity which was zero at the right-hand end and was equivalent to a rotation about the right-hand end of 1 rad/sec. The density was taken as 1 Kg/m.

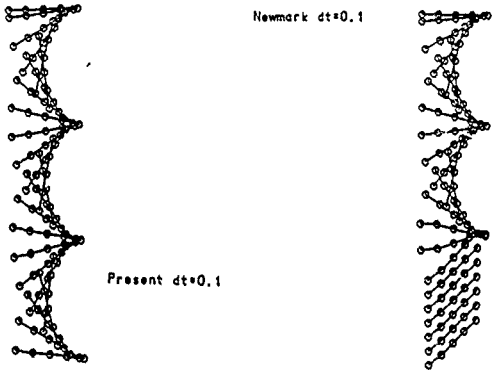


Fig. 7 Deformed configurations for free-flying four-bar chain

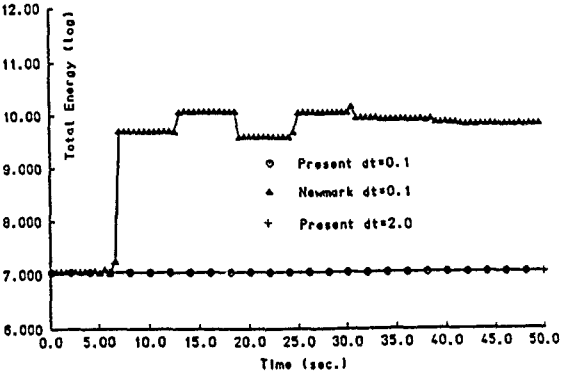


Fig. 8 Relationships between energy and time for free-flying four-bar chain

The response for the present method involves the chain remaining straight as it should In contrast the Newmark method (with $\beta = 1/4$, $\gamma = 1/2$) starts by giving a sensible solution but suddenly jumps to a false energy configuration (Fig. 8) associated with axial oscillation of the bar and no overall rotation (Fig 7)! The solutions in Fig. 7 are related to a fixed time increment of 0.1 secs. They are compared with the exact solution for the horizontal tip-displacement at the left-hand end in Fig. 9 The "exact' solution is computed by assuming that the bars are rigid It can be seen that the present method leads to an effectively exact solution with $\Delta t = 0.1$ while even with $\Delta t = 2.0$, although there is some period elongation, no "locking" occurs. The present method not only conserves energy (Fig. 8) but also angular momentum [14].

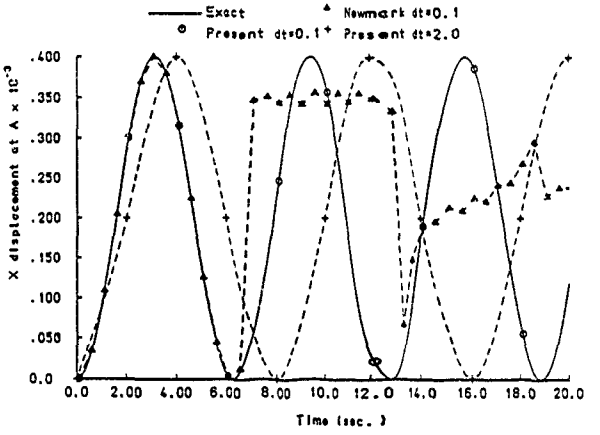


Fig. 9 Relationships between tip-displacement and time for free-flying four-bar chain

4.2 SIMPLE MODEL WITH TWO DEGREES-OF-FREEDOM

The first author and co-worker have described [19,20] an energy framework for the simple model problem shown in Fig 10 The bar OA is modelled with one of the truss elements discussed previously while in the x-direction a linear spring of stiffness K is provided (AB being forced to remain horizontal) The point mass, m, was set to unity , the initial length, ℓ_o, to 100 and Young's modulus times the area of the bar OA was set to 500 so that the stiffness of the bar was 5.0 which is five times larger than the stiffness provided for the linear spring, BA, for which K = 1 No gravity effects were considered

The energy framework can be used to show that if the initial non-dimensional kinetic energy, $\bar{T} = 2T/(K\ell_o^2)$, is greater than the a critical value $\bar{T}_{cr} = 0.833$ the system may, within a certain time period, "escape" from a response associated with a low amplitude so as to respond with a "large amplitude". However, if $\bar{T} < \bar{T}_{cr}$ such "escaping motion" should be impossible For the present study, $\bar{T}_{cr}$ has been set to 0 8 with the angle α in Fig. 10 being 135o.

Figure 11 shows the computed relationships between the the vertical displacement (y direction - Fig. 10) and time. As anticipated, with the present method, the motion does not "escape" (Figs. 11 and 12a). In contrast, with the "Newmark method", escape occurs after about 30 seconds (Figs. 11 and 12b). The latter, false response, is very similar to the true response which might have been obtained with an initial kinetic energy, $\bar{T} > \bar{T}_{cr}$ (see [19,20]).

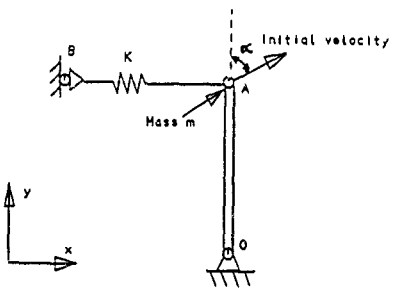


Fig. 10 Simple model with two degrees-of-freedom

5. Extensions to Other Types of Element

The extension of the previous formulation to cover three-dimensional truss elements is trivial Indeed the basic equations remain unaltered. In the following, we will initially outline a possible extension to cover two-dimensional beams

The equivalent static formulation is detailed by the author in [6] and is closely related to the work of Oran [21]. It is based on a Kirchhoff beam formulation without shear deformation although an equivalent formulation based on Timoshenko theory can easily be derived [6] In relation to Fig. 5, we again have two nodes but now with a rotational variable, θ added to the two translational variables, d at each node Collectively the structural "displacements" are then

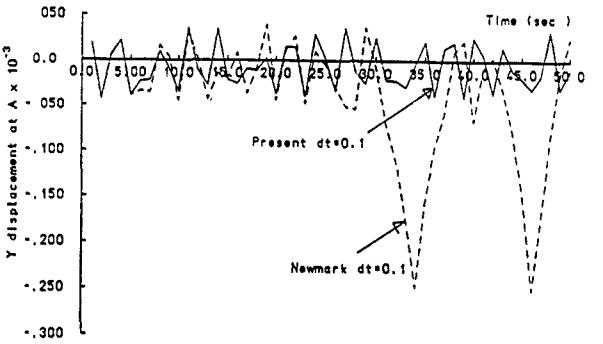


Fig. 11 Relationships between vertical displacement (y) and time for simple model

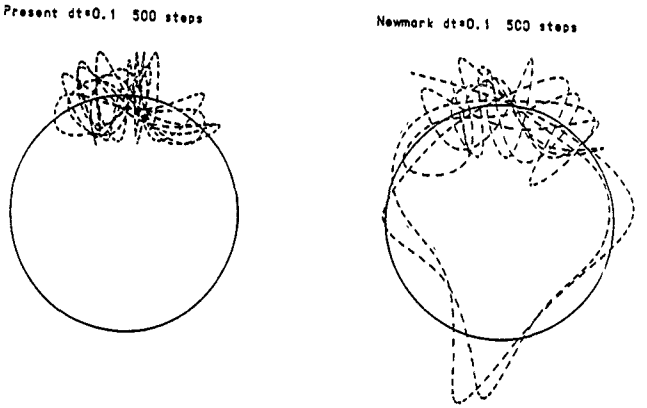


Fig. 12 Traces of motion of point A for simple model
a) Present solution b) Solution via "Newmark"

$$p^T = (d_1^T, \theta_1, d_2, \theta_2) \qquad (31)$$

For the bending deformations, we require two local rotations,

$$\theta_{l1} = \theta_1 - \alpha \quad ; \quad \theta_{l2} = \theta_2 - \alpha \qquad .(32)$$

where α is the rigid body rotation which at configuration n+1 can be computed from:

$$\sin\alpha = \frac{1}{l_0 l_{n+1}} x_{21,o}{}^{x}\, x_{21,n+1} \qquad ..(33)$$

where the lengths l_o (initial) and l_{n+1} (current) follow the same equations as used for the truss element (see (11)). It should be noted that the curved length component is neglected in this basic formulation. It is perfectly possible to add additional terms to allow for these effects [6] via rotating "shallow arch terms".

The local bending moments, $\tilde{M}_1$ and $\tilde{M}_2$ are computed from the local θ's via:

$$\begin{bmatrix}\tilde{M}_1\\ \tilde{M}_2\end{bmatrix} = \frac{EI}{l_o}\begin{bmatrix}4 & 2\\ 2 & 4\end{bmatrix}\begin{bmatrix}\theta_{l1}\\ \theta_{l2}\end{bmatrix} \qquad .(34)$$

For the following dynamic formulation, we require a good approximation to (33) that is valid for the change of rigid body rotation between configurations n and n+1. Such an approximation follows from Figure 13 and is given by:

$$\Delta\alpha = \frac{1}{2}\left[\frac{1}{l_n} f_n + \frac{1}{l_{n+1}} f_{n+1}\right]^T \Delta d_{21} \qquad .(35)$$

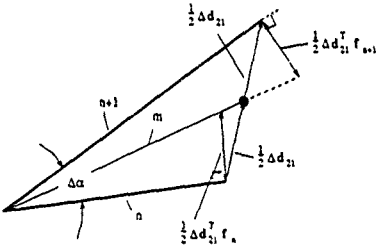


Fig 13 Computation of the rigid-body rotation

The axial energy for the beam element follows identically that given previously for

the truss and hence we will concentrate on the bending energy For the latter, the static internal force vector is given by [6]

$$q_{l,n+1} = B_{n+1}^{T} \begin{bmatrix} \bar{M}_1 \\ \bar{M}_2 \end{bmatrix} \tag{36}$$

where the "B matrix" is given by

$$B_{n+1}^{T} = \begin{bmatrix} \frac{1}{\ell_{n+1}} \begin{bmatrix} -f_{n+1} \\ \ell_{n+1} \\ f_{n+1} \\ 0 \end{bmatrix} & \frac{1}{\ell_{n+1}} \begin{bmatrix} -f_{n+1} \\ 0 \\ f_{n+1} \\ \ell_{n+1} \end{bmatrix} \end{bmatrix} \tag{37}$$

If we now adopt the average B matrix approach of Simo and Tarnow, we arrive at the following mid-point dynamic equilibrium equations

$$g_m = N_{av} \frac{1}{2}\left[b_n + b_{n+1}\right] + \frac{1}{2}\left[B_n + B_{n+1}\right]^{T} \begin{bmatrix} \bar{M}_{av,1} \\ \bar{M}_{av,2} \end{bmatrix}$$

$$+ \frac{1}{\Delta t} \begin{bmatrix} m_1 \Delta v_1 \\ \bar{m}_1 \Delta\theta_1 \\ m_2 \Delta v_2 \\ \bar{m}_2 \Delta\theta_2 \end{bmatrix} - q_{e,m} = 0 \tag{38}$$

where we have also adopted the average B approach for the axial part (see earlier section 3.1) so that the b vectors would be as given in (29) The $\bar{m}$ terms in (38) involve the "lumped" rotational mass while the term $\bar{M}_{av,1}$ is simply the average of the $\bar{M}$ values from configurations n and n+1 (following N_{av} - see (16)).

The time integration of the rotational variables follows the same form as that adopted for the translations, i.e.,

$$\theta_{n+1} = \frac{2}{\Delta t}\Delta\theta - \theta_n \tag{39}$$

$$\theta_m = \frac{1}{\Delta t}\Delta\theta \tag{40}$$

To consider the energy conservation, we firstly supplement the axial strain energy of (19) with the bending strain energy

$$\Delta\varphi_b = \bar{M}_{av,1}\Delta\theta_{f1} + \bar{M}_{av,2}\Delta\theta_{f2}$$

$$= \bar{M}_{av,1}(\Delta\theta_1 - \Delta\alpha) + \bar{M}_{av,2}(\Delta\theta_2 - \Delta\alpha) \qquad (41)$$

where the rigid rotation $\Delta\alpha$ is given by (35). In addition, for any node, we have a contribution to the kinetic energy of:

$$\Delta T_b = \frac{1}{2}\bar{m}(\dot{\theta}_n + \dot{\theta}_{n+1})\Delta\theta = \frac{\bar{m}}{2\Delta t}\Delta\theta\Delta\theta \qquad (42)$$

where we have used (39) Combining (41), (42) and (35) with (14), (15) and (19) leads to the relationship:

$$\Delta\sigma = \Delta\varphi + \Delta T + \Delta P = g_m^T \Delta p \qquad (43)$$

where g_m is the residual term given by (38) and Δp contains the incremental nodal "displacements". As with the truss formulation the change of energy in (43) will therefore be zero once we have iterated to the state where the residual in (38) is zero

In relation to static analysis, the author has extended the two-dimensional co-rotational formulation to three-dimensional beams [22] and, in future work, it is intended to apply a similar approach to the dynamic case. The extension to dynamics will also be applied to a simple co-rotational triangular shell element [13]. If the formulation is restricted to small strains (although with large deformations and rotations), the simplest approach would be to adopt the method based on the averaged **B** matrix. For the shell element of [13], which has only one rotational variable per each mid-side, the resulting formulation should be very simple.

6. Conclusions

An energy conserving time-integration procedure has been devised that is based on the co-rotational approach and involves a mid-point configuration in conjunction with averaged strains Examples have been presented showing the application of the technique to simple two-dimensional truss elements. It has been shown that "conventional techniques" lead to solution that may suddenly jump to higher energy states that are associated with high frequency axial oscillations Alternatively, they may exhibit false "escaping motion" that should be associated with a structure subject to a higher initial kinetic energy In contrast, the current solutions remain stable with large time-steps

A disadvantage of the method is that it involves a non-symmetric effective tangent stiffness matrix However, this disadvantage should be more than compensated by the stability of the method. In addition, it is possible to artificially symmetrise the matrix although convergence difficulties sometimes result

Although the method has so far only been applied to simple elements, the paper has indicated possible techniques for extending the procedure to other elements such as three-dimensional beams and shells that can be formulated using the co-rotational method

7. References

1. Stewart, I., Warning - handle with care!, Nature, 355, 1992, pp.16-17.
2. Downer, J D., Park, K.C & Chiou, J C., Dynamics of flexible beams for multibody systems: a computational procedure, Comp Meth in Appl. Mech & Engng, 96, 1992, pp. 373-408
3. Belytschko, T., An overview of semidiscretisation and time integration procedures in: Computational methods for transient analysis, ed T Belytschko & T.J.R. Hughes, Elsevier Science, 1983, pp.1-65.
4. Bathe, K-J, Finite element procedures in engineering analysis, Prentice-Hall, Englewood Cliffs, 1982.
5. Zienkiewicz, O.C., & Taylor, R.L., The finite element method, Vol. 2, Fourth Ed., McGraw-Hill, Maidenhead, 1991
6. Crisfield, M.A., Non-linear finite element analysis of solids and structures, Vol. 1, J. Wiley, Chichester, 1991
7. Haug, E., Nguyen, O S. & De Rouvray, A.L., An improved energy conserving implicit time integration algorithm for nonlinear dynamic structural analysis, Proc. 4th Int. Conf on Struct Mech. in Reactor Tech., San Francisco, Aug. 1977.
8. Hughes, T.J.R., Liu, W.K. & Caughy, Transient finite element formulations that preserve energy, J. Appl. Mech., 45, 1978, 366-370.
9. Simo, J.C. & Wong, K.K., Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum, Int. J. for Num. Meth. in Engng., 31, 1991, 19-52.
10. Simo, J.C., Rifai, M.S. & Fox, D.D., On a stress resultant geometrically exact shell model, Part VI: Conserving algorithms for non-linear dynamics, Int. J. for Num Meth. in Engng., 34, 1992, 1117-164
11. Hilber, H.M., Hughes, T.J.R. & Taylor, R.L., Improved numerical dissipation for time integration algorithms, Int. J. of Earthquake Engng. & Struct. Dynam., 5, 1977, 283-292.0
12. Zienkiewicz, O.C., Wood, W L & Taylor, R.L., An alternative single-step algorithm for dynamic problems.J. of Earthquake Engng. & Struct. Dynam., 8, 1980, 31-40
13. Peng, X. & Crisfield, M.A., A consistent co-rotational formulation for shells using triangular elements with constant strain and constant curvature, Int. J. for Num. Meth. in Engng., 35, 199, 1829-1847.
14. Crisfield, M.A. & J. Shi, A co-rotational element/time integration strategy for non-linear dynamics, submitted to IJNME, 1992.
15. Crisfield, M.A., Shi, J. & Lim, K.L., Finite elements and non-linear dynamics, in: Modern Practice in Stress and Vibration Analysis, The Stress Analysis Group of the Institute of Physics, Sheffield, April, 1993.
16. Simo, J C. & Tarnow, N., The discrete energy-momentum method Conserving algorithms for nonlinear elastodynamics, Z. angnew Math Phys., 43, 1992, 757-792
17. Simo, J C & Tarnow, N., A new energy conserving algorithm for the nonlinear dynamics of shells, IJNME to be published.
18. Bathe, K-J, Finite element procedures for solids and structures - nonlinear analysis, MIT Centre for Advanced Engineering Studies, 1986.
19. Lim, K L & Crisfield, M A., Dynamic finite element analysis applied to a simple model exhibiting dynamic instability, submitted to Engineering Computations
20. Lim, K L., Dynamic Stability of suddenly loaded structures, M.Sc Thesis,

Dept of Aeronautics, Imperial College, London, 1992
21. Oran, C., Tangent stiffness in plane frames, ASCE, J Struct Div, 99, 1973, 973–985.
22 Crisfield, M.A., A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements, Comp. Meth in Appl Mech & Engng, 81, 1990, 131–150

MODIFICATION OF FLEXIBLE BODY DYNAMICS USING SUBSTRUCTURES

STANLEY G. HUTTON and MALCOLM J. SMITH
Department of Mechanical Engineering
University of British Columbia
Vancouver, B.C. Canada V6T 1Z4

ABSTRACT. The calculation and modification of the undamped frequency characteristics of finite element models is considered in the context of component mode synthesis procedures. The application of the free interface component mode procedure is examined by analysis of a complex 3D model. Developments are presented that allow significantly reduced models to be utilized in the modification of global frequency characteristics. The application of sensitivity, forward modification and inverse modification approaches are examined.

1. Introduction

Analysis of the dynamics of flexible mechanical systems is generally achieved by the use of the finite element method. Such an approach often leads to the development of a mathematical model with a large number of degrees of freedom. In many cases, the dynamics of interest only involve the lower modes of vibration and in such cases the application of component mode synthesis principles to the model may be advantageous. In this approach, the overall model is viewed as a series of components with various methods having been proposed for idealizing the behaviour of each component. The overall aim of the procedure is to define component modes, that serve as basis vectors for the displacements of each component, and a synthesis procedure that can be used to couple the component equations into a global representation of the system being modeled. The benefit of this procedure is that the resulting equations contain far fewer variables than the original model, but by proper choice of component representation, the reduced model can accurately predict the lower frequency characteristics of the original system. Further, should design modifications be required, only the characteristics of specific components may need to be changed and substantial time saving can be effected in the re-analysis process. Such procedures thus allow the possibility of more design alternatives to be reviewed in a given time, thus increasing the chances of an optimal design. In this context it should be remarked that the availability of relatively fast procedures that assist in the design stage, even though they may sacrifice some accuracy, can be very useful, given the fact that the analysts can always resort to a detailed finite element model as a final check. It is with this philosophy in mind that some of the developments in the paper are presented.

In the present study, attention is confined to the determination of the modal characteristics of linear, undamped, non-gyroscopic vibrating systems. In this context, the purpose is to illustrate how

component mode synthesis procedures can be utilized in the analysis and modification of finite element models.

2. Component Mode Synthesis

The problem to be considered is the determination of the free vibration characteristics of a system that can be represented by a global equation of motion of the form

$$Mu(t) + Ku(t) = f(t) \qquad (1)$$

where M and K are real symmetric mass and stiffness matrices, u is a vector of time dependent generalized displacements and f a vector of applied forces. In general, M is positive definite and K is either positive definite or positive semi-definite, depending upon whether rigid body motion is possible.

The philosophy of component mode synthesis is to construct an alternative global equation that is determined on the basis of modal and response characteristics that are determined for individual sub-structures or components of the global system. Reviews of component mode representation have been given by Hurty [1], Kubomura [2] and Craig [3]. Smith [4] has presented an evaluation of the applicability of such methods for modal analysis of finite element models.

The free vibration equations of motion formulated by component mode synthesis in its most general form may be written as:

$$\hat{K}\,\xi - \lambda\,\hat{M}\,\xi = 0 \qquad (2)$$

where

$$\hat{K} = T^T \overline{K}\, T \qquad \textit{and} \qquad \hat{M} = T^T \overline{M}\, T$$

T = the transformation from the physical coordinates of the components u to the generalized coordinates of the system ξ . T thus depends upon the CMS procedure used.

$\overline{K}$, $\overline{M}$ = block diagonal matrices containing the stiffness (mass) matrix of each component written in terms of physical coordinates u.

In this paper a free interface method is used in which components are represented by low frequency free vibration modes and residual flexibilities of the neglected high frequency modes. For this method:

$$T = \overline{\Phi} - \overline{\Psi}\; A[A^T \overline{\Psi}^B A]^{-1} A^T \overline{\Phi}^B; \qquad \xi = \bar{p} \qquad (3)$$

where

$\overline{\Phi}$ = block diagonal matrix each component of which contains the free free modes used in that component representation

$\overline{\Phi}^B$ = block diagonal matrix each component of which contains the components of the free free modes used that correspond to the interface location of that component.

$\overline{\Psi}$ = block diagonal matrix each component of which contains the residual attachment modes used in that component representation

$\overline{\Psi}^B$ = similar to $\overline{\Phi}^B$

A = matrix defining intercomponent compatibility and equilibrium constraint.

$\overline{p}$ = generalized coordinates of the global model containing in sequence the magnitude of the contribution of each component free mode to the global response.

As an example of the use of such CMS procedures, consider the analysis of the structure shown in Figure 1. The structure in question consists of a telescope focus unit that is supported above the main mirror by latticed trusses. A detailed analysis of this structure was undertaken by Smith [4] using both free interface and fixed interface CMS formulations. The complete system was analysed in terms of five components that described the body of the focus unit and four components representing each of the legs. Table 1 summarizes the results of three different analysis using the same component modes but where the highest frequency retained in any component was respectively 30, 90 and 150 Hz. Table 1 also compares the resulting calculated natural frequencies for the lowest 40 modes of the system for each different case with the results of a global finite element model. Also compared are the number of degrees of freedom that are involved in each representation.

The first point to note is the degree of condensation that is obtained by the procedure, the number of degrees of freedom are reduced from 3360 to either 46, 62 or 78 depending upon the component mode frequency cut-off used.

The CMS results for the 30 Hz component cut-off predict to an accuracy greater than 5% all global modes below 33.5 Hz, for 90 Hz cut-off, all global modes below 98 Hz, and for 150 Hz cut-off the lowest 40 frequencies are calculated to less than 1% error.

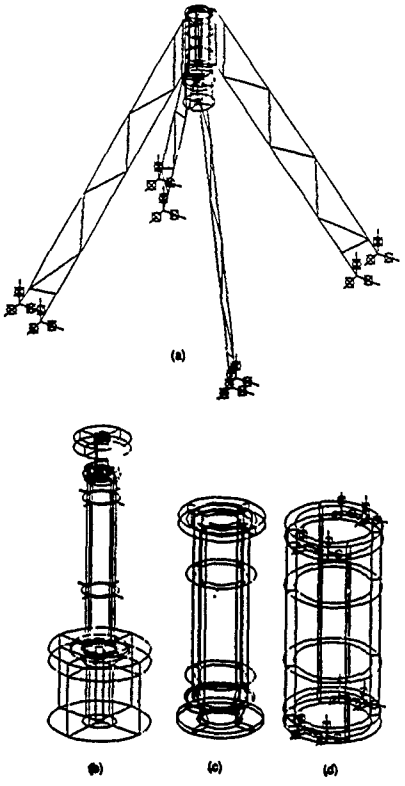


Figure 1 Model of Telescope Focus Unit and Supporting Trusses
(a) General Arrangement (b) Inner Tube with Screw Assembly Attached to Top and Chopping Mechanism to Bottom (c) Outer Tube (d) Support Tube

TABLE 1: Natural frequency results for the telescope model

Direct F.E.M. Analysis		% Error, Free-Interface CMS Analysis		
		Maximum Component Frequency		
Mode	*Frequency*	*30 Hz*	*90 Hz*	*130 Hz*
1	11.5	0 02	0 00	0 00
2	13.3	0.05	-0.01	-0 01
3	13.3	0.26	0.02	0.02
4	13.4	-0.03	-0 02	-0 02
5	25.5	2.93	0.01	0.01
6	33.5	6 87	0.03	-0 04
7	34 0	6 55	-0.06	0.08
8	34.6	10 6	0.08	0 03
9	34.6	16.6	-0.01	-0.08
10	39.1	7.08	0 03	0.00
11	42.2	2.09	0.06	0.03
12	45.3	4.88	-0.03	0 04
13	45.3	8.29	-0.02	0.01
14	45 3	9 95	0.14	0.03
15	45 6	9.33	-0.02	-0 02
16	50 6	46.1	-0.03	-0 06
17	61.0	30.1	0.07	-0.10
18	61.0	42.5	0.07	-0 06
19	61.1	72.1	0 07	-0 01
20	71.0	54 7	0.89	0 08
21	73.1	43.4	0.38	0 05
22	83.7	67.7	0.18	0 01

table continued

TABLE 1, continued

Mode	Frequency	30 Hz	90 Hz	130 Hz
23	83 9	70 0	0 60	-0.10
24	84 0	206	0.75	0 04
25	85 6	225	1.38	0.04
26	85.7	232	2 83	-0.02
27	98.1	375	7.21	0.02
28	99.0	391	9 57	0.25
29	99.7	537	9.63	0 19
30	104 0	578	8.10	0.03
31	104.1	670	12.0	0 24
32	104.3	943	12 2	0.31
33	105.9	1018	10 6	0.07
34	117.3	964	19 4	0 22
35	117.5	1000	19 4	0.29
36	117.7	1043	52.3	0 38
37	120.1	1072	49.7	0 43
38	125.5	1334	44 6	0.10
39	127.4	1388	78 5	0.97
40	133 6	1453	92.6	0.00
# dof	**3360**	**46**	**62**	**78**

Thus, using CMS procedures, it is possible to significantly reduce the number of degrees of freedom as well as subdivide the global structure into substructures that may allow future modifications to be accomplished more efficiently. For example, if the required modification only involves specific components, then it is not necessary to recompute the characteristics of the unmodified components. This can lead to substantial time savings. In the telescope focus unit analysis, the effect of changing the stiffness of the bearings between the inner and outer tubes of the focus unit was investigated. Modification of this stiffness and re-analysis of the complete problem took 22% of the time required for the baseline CMS analysis. Thus in situations where many such re-analyses have to be computed in order to obtain a system with the desired response characteristics, the benefits of using a CMS procedure may be significant.

3. Frequency Modification Procedures

When faced with the task of modifying the vibration characteristics of a system to meet specified response criteria, often it is the location of system natural frequencies with respect to applied forcing frequencies that is of interest. Thus a common requirement is that of modifying modal characteristics in some prescribed manner. Procedures available for the purpose have been reviewed by Baldwin and Hutton [5]. So-called "forward modification procedures" involve specifying a physical change to the model and computing the resulting modal effect. Such procedures may require a number of iterations to achieve a satisfactory modification. Inverse modification procedures, on the other hand, specify as input the required modal modifications and the output from the algorithm is the required modifications to the system. Such modification should be expressed in terms of parameters that can be physically changed such as plate thicknesses, or beam second moments of area as opposed to the elements of the mass and stiffness matrices which then require further interpretation into appropriate physical variables.

In the case of finite element analysis, it is convenient to define modification of the stiffness matrix, for example, as

$$\Delta \overline{K} = \sum_{e} \sum_{j=1}^{m} A_e^T (k_e)_j A_e \alpha_j \qquad (4)$$

where the modification of the j^{th} design parameter results in a change in the e^{th} element stiffness of $(k_e)_j \alpha_j$ The matrix A_e represents the transformation between element and global coordinates in the finite element model. A similar equation can, of course, be written for mass matrix modification in terms of element mass sensitivity matrices $(m_e)_j$. If the element stiffness and mass sensitivity matrices are independent of the design parameter α, then the modification is termed linear; changing mass density or Young's Modulus results in linear modifications. Changing the thickness of a plate is a linear change for a membrane element but a non-linear change for plate bending element. It is possible to approximate non-linear modifications as linear if the change in the design parameter is sufficiently small. In the present work, all changes will be assumed linear.

3.1 SENSITIVITY ANALYSIS

General discussions of sensitivity methods in structural dynamics have been presented by Adelman and Haftka [6], and Brandon [7]. The basic sensitivity equations were derived originally by Lancaster [8], and Fox and Kapoor [9].

The sensitivity of the i^{th} eigenvalue to a change in the design parameter r is given by

$$\frac{\partial \lambda_i}{\partial r} = \xi_i^T \left[\frac{\partial \tilde{K}}{\partial r} - \lambda_i \frac{\partial \tilde{M}}{\partial r} \right] \xi_i \qquad (5)$$

where $\tilde{K} = T^T \overline{K}^T T$, $\tilde{M} = T^T \overline{M}\ T$ and $\xi_i^T T^T \overline{M}\ T\ \xi_i = 1$. In general, the rate of change of T with respect to r cannot be evaluated directly, and thus this formula cannot be used directly for CMS models. However, it is beneficial to consider circumstances in which T can be considered invariant with respect to r. Changes linearly proportional to the global mass matrix $\overline{M}$ fall into this category when using the free interface method. Under these conditions, equation (5) can be written as

$$\frac{\partial \lambda_i}{\partial r} = -\lambda_i \xi_i^T T^T \frac{\partial \overline{M}}{\partial r} T \xi_i \qquad (6)$$

Unfortunately equation (6) is of limited value in that it is in the sensitivity of the frequency to a local change in a component that is of interest. Assume that T can be considered invariant with respect to r, then equation (5) can be written:

$$\frac{\partial \lambda_i}{\partial r} = \xi_i^T T^T \left[\frac{\partial \overline{K}}{\partial r} - \lambda_i \frac{\partial \overline{M}}{\partial r} \right] T \xi_i \qquad (7)$$

If T and $\overline{K}$ and $\overline{M}$ are written in terms of their component contributions

$$\frac{\partial \lambda_i}{\partial r} = \sum_{j=1}^{N} \xi_i^T T_j^T \left[\frac{\partial \overline{K_j}}{\partial r} - \lambda_i \frac{\partial \overline{M_j}}{\partial r} \right] T_j \xi_i \qquad (8)$$

where $T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_N \end{bmatrix}$

$\overline{K} =$ *diagonal* $[K_1, K_2 \ldots K_N]$

and N = total number of components.

Thus, equation (8) provides information as to which component is most sensitive in modifying λ_i when design parameter r is changed. Although no mathematical proof of the validity of equation (7) has been presented, under some circumstances it may be a useful approximation. For example, if the component free vibration and residual attachment modes are independent of r, then (7) is valid. Thus it may be assumed that for changes that do not violate this condition too greatly, the resulting

sensitivities will have some validity. In general, the use of equation (8) should be confined to investigating which component is a candidate for modification rather than attempting to use the calculated sensitivity to predict the required change in the design parameters.

As an example of the use of equation (8), the sensitivities of the 10th and 11th natural frequencies of the model shown in Figure 1 are evaluated with respect to changes in density and Young's Modulus. The results, which are shown in Table 2, indicate that modification of these modes is most effectively done by changing the mass of the chopping mechanism and the Young's Modulus of the inner tube. In subsequent section, a more exact procedure will be utilized to determine the magnitude of the modifications required in order to produce a specified change in frequency.

TABLE 2: Sensitivity of Eigenvalues to Component Densities and Young's Modulus

Component	$\frac{\rho}{\lambda_{10}}\frac{\partial\lambda_{10}}{\partial\rho}$	$\frac{E}{\lambda_{10}}\frac{\partial\lambda_{10}}{\partial E}$	$\frac{\rho}{\lambda_{11}}\frac{\partial\lambda_{11}}{\partial\rho}$	$\frac{E}{\lambda_{11}}\frac{\partial\lambda_{11}}{\partial E}$
Inner Tube	-.053	.111	-.053	.139
Outer Tube	- 041	.024	-.039	.027
Screw Assembly	-.002	.003	- 001	.003
Chopping Mechanism	- 424	.002	- 457	003
Support Tube	- 062	.005	-.058	.002

3.2 FORWARD MODIFICATION

The global equation of motion for the CMS model is given in equation (2) and when written in terms of the assembled component stiffness $\overline{K}$ and mass matrix $\overline{M}$ is, for the i^{th} mode

$$T^T \overline{K}\ T\xi_i = \lambda_i T^T \overline{M}\ T\xi_i \qquad (9)$$

where the i^{th} mode shape defined in assembled physical coordinates $\overline{u}_i$ is given by

$$\overline{u}_i = T\xi_i \qquad (10)$$

Consider now the effect of modifying the stiffness and mass characteristics of this model. The task is to determine the new mode shapes $\overline{u}_i'$ without having to recalculate T and conduct a complete

analysis of the modal. Assume that the new modes in physical coordinates can be sufficiently well approximated by a linear combination of the calculated baseline modes, i.e.,

$$\overline{u}_i^{'} = \sum_{j=1}^{p} c_{ij}\overline{u}_j = \sum_{j=1}^{p} c_{ij}T\overline{\xi}_j = TZc_i \qquad (11)$$

where Z is a matrix of the baseline modes ξ_i, and c_i is a vector of constants to be determined. Under the constraints imposed by (11), the resulting equations of motion for a new system with modifications defined by $\Delta\overline{K}$ and $\Delta\overline{M}$ are given by

$$T^T\left[\overline{K} + \Delta\overline{K} - \lambda_i^{'}\left(\overline{M} + \Delta\overline{M}\right)\right]T\xi_i^{''} = 0 \qquad (12)$$

where the modified mode

$$\overline{u}_i^{'} = TZc_i = T\xi_i^{''}$$

Once T has been calculated, equation (12) affords a convenient and rapid approach to determining the effect of arbitrary changes $\Delta\overline{K}$ and $\Delta\overline{M}$ to the baseline modes.

Using the forward modification approaches outlined, it is possible to significantly reduce the order of complex finite element models. Although in some cases the absolute accuracy of the analysis may not be known, the ability to investigate a larger number of modification alternatives may be more important. A final full finite element re-analysis would generally be conducted in such cases to verify the accuracy of condensations adopted.

As an illustration of the use of equation (12), consider the effect of changing the material of the component representing the chopping mechanism from aluminum to steel. As this component is effectively rigid, this change is essentially a pure mass change. Figure 2a shows the results. The frequencies of the aluminum baseline modes are shown, as are the modified frequencies calculated by equation (12) and referred to as the linear equivalent analysis. A subsequent complete CMS analysis of the steel case gives exactly the same results as the linear equivalent analysis.

Figure 2b shows the results of a second analysis in which the modification again involves changing the material, for the component representing the inner tube, this time from steel to aluminum. In this case, the modification affects the dynamics both through the stiffness and the mass matrices. As may be seen, the linear equivalent re-analysis gives results close to those produced by a complete CMS re-analysis.

In both examples, the use of the linear equivalent analysis results in considerable time savings compared to a CMS re-analysis.

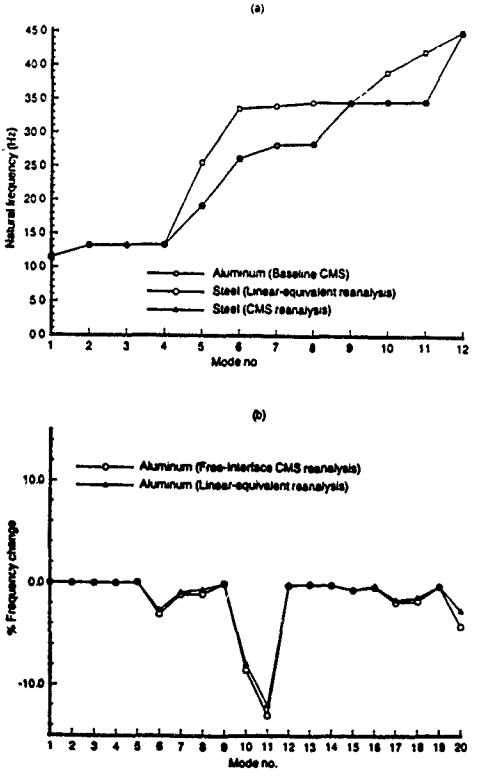


Figure 2 Modification of Natural Frequencies Using Linear Equivalent Analysis (a) Chopping Mechanism (b) Inner Tube Modification

3.3 INVERSE MODIFICATION

Inverse Modification techniques based on linearized perturbation equations were first developed by Stetson [10]. Subsequently, Hoff *et al* [11] developed a predictor-corrector algorithm using both linear and general perturbation equations for unsubstructured systems. In a recent paper by Smith and Hutton [12], a new formulation of the perturbation equations for inverse frequency modification is presented which retains all non-linear coupling terms and is effective for large frequency changes. The following discusses the application of this formulation to CMS models.

Assume that it is required to modify the dynamics of a model such that $\lambda_i = \lambda_i^*$ by varying a design parameter where the fractional change in this parameter is denoted by α.

In general, the transformation T will be an unknown function of the design change α. However, by computing all the baseline modes and utilizing equation (11), one can write

$$\left(\Lambda + Z^T T^T \Delta \overline{K}(\alpha) TZ\right) c_i = \lambda_i^* \left(I + Z^T T^T \Delta \overline{M}(\alpha) TZ\right) c_i \qquad (13)$$

where Λ is a diagonal matrix of the baseline eigenvalues and the Z matrix has been normalized such that $Z^T T^T \overline{M}\, TZ = I$

If $\Delta \overline{K}$ and $\Delta \overline{M}$ are restricted to be linear functions of α and there are m such design variables, then equation (13) may be written

$$\sum_{j=1}^{m} \alpha_j Z^T T^T \left(\overline{K_j}^* - \lambda_i^* \overline{M_j}^*\right) TZ c_i = (\Lambda - \lambda_i^* I) c_i \qquad (14)$$

where $\overline{K_j}^*$, $\overline{M_j}^*$ are global matrices representing the stiffness and mass sensitivities to the design variable α_j. Equation (14) is thus a system of equations whose order is equal to that of the CMS model. If there are multiple frequency constraints then there are as many sets of (14) as there are frequency constraints. This equation has the same form as that for unsubstructured models and can be solved in the same manner.

If m is equal to one and there is only one frequency constraint, an eigenvalue problem results with the eigenvalues representing the change α_j and the eigenvector c_i determining, through the use of (11), the modified eigenvector u_i. In the more general case when the number of design variables exceeds the number of frequency constraints, an iterative solution may be used [12].

As an application of equation (14), the fractional change in the density of the chopping mechanism of the model was calculated in order to reduce the frequency of the 10th mode from 39 Hz to 35 Hz. The resulting value of α was 1.37, i.e., the density should be increased by a factor of 2.37. Using the same frequency specification but allowing two design values α_1, corresponding to the density of the

chopping mechanism, and α_2, corresponding to the Young's Modulus of the inner tube, gave values of $\alpha_1 = 1.07$ and $\alpha_2 = -0.47$. Table 3 presents a comparison of the baseline frequencies and the frequencies resulting from the modifications indicated and shows that the required frequency shift has been obtained.

It may be noted that in the case of the mass modification, the magnitude of the density change is much larger than would have been expected from the sensitivity result quoted in Table 2. This is due to the fact that the sensitivity calculated at 39 Hz is not valid at 35 Hz. Recalculation of the sensitivity of the chopping mechanism of 35 Hz gives $\frac{\rho}{\lambda_{10}} \frac{\partial \lambda_{10}}{\partial \rho} = -.070$. Thus the mode shapes have been so modified as to change significantly the effect of different design parameters upon the system frequencies. This result emphasizes the limited value of the sensitivity analysis.

TABLE 3: Modification of Tenth Natural Frequency Using Perturbation Analysis

Baseline		Density Modification		Density and Modulus Modification	
Mode	*Frequency*	*Modified Frequency*	*% Change*	*Modified Frequency*	*% Change*
1	11.53	11.46	-0 6	11.48	-0 5
2	13 27	13 27	-0 0	13.27	-0 0
3	13 28	13 27	-0.0	13.27	-0 0
4	13.35	13 35	0.0	13 35	0 0
5	25.55	20.79	-18 6	21.61	-15 4
6	33 55	28 31	-15 6	27.14	-19.1
7	34 00	28.83	-15.1	28.85	-15.1
8	34.59	30.48	-11.9	28 98	-16 2
9	34 63	34 64	0 0	34 60	-0 1
10	39 09	34.93	-10.6	34.70	-11 2
11	42 18	35 00	-17.0	34 71	-17 7
12	45.30	45.21	-0.2	45 20	-0 2

4. Summary

In the design of flexible structure, especially those that are modeled with large order finite element representations, the availability of analysis procedures that can provide rapid evaluation of different design modifications is desirable Re-running the baseline finite element model is not economical.

In the present paper, procedures have been presented which utilize the principles of component mode synthesis to provide the designer with the ability to efficiently assess the effect of prescribed modifications or to compute the design changes required in order to achieve a desired frequency modification in the structure. Examples have been presented for a complex three-dimensional finite element model that illustrates the efficiency of the methods described.

5. References

1. Hurty, W. C., "Introduction to Modal Synthesis Techniques", Synthesis of Vibrating Systems, ASME booklet, Nov. 1971, pp. 1--13.

2. Kubomura, K., "A Theory of Substructure Modal Synthesis.", Journal of Applied Mechanics, Vol. 49, No. 12, December 1982, pp. 903--909.

3. Craig, R. R., "A Review of Time-Domian and Frequency-Domain Component-Mode Synthesis Methods", International Journal of Analytical and Experimental Modal Analysis, Vol. 2, No. 2, April 1987, pp. 59--72.

4. Smith, M.J., "An Evaluation fo Component Mode Synthesis for Modal Analysis of Finite Element Models", Ph.D. Thesis, Dept. of Mechanical Engineering, University of British Columbia, 1993.

5. Baldwin, J. F., and Hutton, S. G., "Natural Modes of Modified Structures", AIAA Journal, Vol. 23, No. 11, November 1985, pp. 1737--1743.

6. Adelman, H. M., and Haftka, R. T., "Sensitivity Analysis of Discrete Structural Systems", AIAA Journal, Vol. 24, No. 5, May 1986, pp. 823--832.

7 Brandon, J. A., Strategies for Structural Dynamic Modification, Research Studies Press, Somerset, U.K., 1990.

8. Lancaster, P., "On Eigenvalues of Matrices Dependent on a Parameter", Numerische Mathemank, Vol. 6, No. 5, December 1964, pp. 377--387.

9. Fox, R. L., and Kapoor, M. P., "Rates of Change of Eigenvalues and Eigenvectors", AIAA Journal, Vol. 6, No. 12, December 1968, pp. 2426--2429.

10 Stetson, K. A., and Palma, G. E , "Inversion of First-Order Perturbation Theory and Its Application to Structural Design", AIAA Journal, Vol. 14, No 4, April 1976, pp. 454-460

11 Hoff, C. J., Bermitsas, M. M., Sandstrom, R. E., and Anderson, W. J , "Inverse Perturbation Method for Structural Redesign with Frequency and Mode Shape Constraints", AIAA Journal, Vol. 22, No 9, Sept. 1984, pp. 1304--1309.

12. Smith, M. J., and Hutton, S. G., "A Perturbation Method for Inverse Frequency Modification of Discrete, Undamped Systems", Submitted for publication to Journal of Applied Mechanics.

VALIDATION OF FLEXIBLE BEAM ELEMENTS IN DYNAMICS PROGRAMS

J.P. Meijaard
Laboratory for Engineering Mechanics
Delft University of Technology
Mekelweg 2, NL-2628 CD Delft, The Netherlands

Abstract

A spatial beam element for static and dynamic problems which involve large displacements and rotations is described. This beam element is applied to static linear buckling problems, the simulation of the motion of a slider-crank mechanism with a flexible connecting rod and a planar and spatial spin-up motion of a flexible beam. Results are compared with those from the open literature.

1. Introduction

With the increase of the speed of operation and the required accuracy and the tendency towards a lighter construction of machinery such as linkages and manipulators, the influence of deformation on the dynamics of mechanical systems becomes more important. The machine parts can no longer be considered as rigid and flexibilities have to be included in the numerical analysis of the motion. Especially the deformations of slender elements such as beams, plates and shells have a noticeable influence.

In the description of the behaviour of a system, one usually starts from a physical model, which is transformed into a mathematical model, in which some simplifications of the physics are made. The mathematical model is translated into a numerical model by some discretization and solution process. The numerical model has to be implemented in a computer program, after which simulations can be made of the behaviour of the considered system. As exact solutions for non-linear dynamical systems which are relevant to engineering problems are hardly known, the validation of the results which are obtained by simulations constitutes a non-trivial task. The implementation can be checked by comparing the results from different independently developed programs based on the same numerical model. The validity of the numerical model can be checked by comparing results obtained from programs which are based on different discretization or solution methods and sometimes from convergence and stability theorems. The completeness of the mathematical model can be investigated by studying the influence of the inclusion of some effects that were initially left out of consideration; it is, however, difficult to identify the

important effects in advance. A final validation, an answer to the question to what extent the results represent reality, must come from a comparison between results from simulations and measurements on experiments.

In this paper we restrict ourselves to the modelling of uniform elastic beams in which strains remain small, but which may undergo large translations and rotations. As these are the simplest and most widely used slender parts and Kane *et al.* [1] claimed that some existing multibody formulations may give incorrect results, a detailed investigation into the modelling of beams appears to be important. Firstly, a finite element for spatial beams with a doubly symmetric solid cross-sections is described, which is expected to be capable of producing sufficiently accurate results. For a demonstration of the accuracy and the proper representation of the geometric stiffness, the beam element is applied to some spatial linear buckling problems for which exact solutions are known. The much studied dynamic problems of a planar slider-crank mechanism with a flexible connecting rod and a planar and spatial spin-up manoeuvre of a flexible beam are simulated and results are compared with those from the literature. In the last example, it is shown how some aspects of beams with a thin-walled open cross-section, such as constrained warping and the eccentricity of the shear centre, can be approximately included in the analysis.

2. Description of a Beam Element

In this section a spatial beam element is described. For static analysis, the two-dimensional version of this element was developed by Visser and Besseling [2]. An extension to three dimensions and its use in the buckling and post-buckling analysis of frameworks was given by Besseling [3–6]. The mass description was included by Van der Werff and Jonker [7,8] and the author [9]. Here some refinements are added.

The finite beam element is based on the elastic line concept. This means essentially that the beam is slender and the cross-section is doubly symmetric and more or less solid. The configuration of the element is determined by the positions and orientations of the end nodes, by which it can be coupled to and interact with other elements. The positions of the end nodes p and q are given by their coordinates $\mathbf{x}^p$ and $\mathbf{x}^q$ in a global inertial system (OXYZ) and the orientations are given by orthogonal triads of unit vectors, $(\mathbf{e}_x^p, \mathbf{e}_y^p, \mathbf{e}_z^p)$ and $(\mathbf{e}_x^q, \mathbf{e}_y^q, \mathbf{e}_z^q)$, which are rigidly attached to the nodes. $\mathbf{e}_x$ is perpendicular to the average cross-sectional plane of the beam in the sense of Cowper [10] and $\mathbf{e}_y$ and $\mathbf{e}_z$ are in the directions of the principal axes of the cross-section. If transverse shear deformations can be neglected, $\mathbf{e}_x$ is tangent to the central axis (elastic line) of the beam. The orientations can be described by several choices of parameters such as Euler angles, modified Euler angles, Rodrigues parameters and Euler parameters. In the ultimate calculations we shall use Euler parameters, but this choice is immaterial to the description of the properties of the element.

The element has six degrees of freedom as a rigid body, while the nodes have twelve degrees of freedom. Hence the deformation that is determined by the end nodes of the element can be described by six independent generalized strains, which are functions of the positions and orientations of the nodes. If the parameters which describe these positions and orientations are grouped

together in a vector x, and the vector of deformations is denoted by $\bar{\varepsilon}$ the deformation map can be written symbolically as

$$\bar{\varepsilon}_i = D_i(x_k). \tag{1}$$

With $l = x^q - x^p$ and l_0 the length of the undeformed beam, we define the six generalized strains as

$$\bar{\varepsilon}_1 = \|l\| - l_0, \qquad \bar{\varepsilon}_2 = l_0(e_z^p.e_y^q - e_y^p.e_z^q)/2,$$
$$\bar{\varepsilon}_3 = -l.e_z^p, \qquad \bar{\varepsilon}_4 = l.e_z^q,$$
$$\bar{\varepsilon}_5 = l.e_y^p, \qquad \bar{\varepsilon}_6 = -l.e_y^q. \tag{2}$$

These generalized strains, which may be compared to what Argyris calls natural modes [11], are invariant under arbitrary rigid body motions or orthogonal coordinate transformations, so they truly measure the amount of strain of the element.

The dual quantities of $\bar{\varepsilon}$ are the generalized stresses $\bar{\sigma}$. They are defined by the property that $\bar{\sigma}.\delta\bar{\varepsilon}$ represents the internal virtual work done by the stresses in the element due to the virtual deformations $\delta\bar{\varepsilon}$. In the case of small deformations, the generalized stresses have a clear physical meaning. As the deformed and undeformed geometry are nearly the same, we consider the undeformed situation in which the beam central axis coincides with the global X-axis, and an equilibrium force system given by the nodal forces F^p and F^q and the nodal moments M^p and M^q acts on the element. With $\lambda = (F^p, M^p, F^q, M^q)$ and the small nodal displacements and rotations dual to the forces and moments $u = (u^p, \varphi^p, u^q, \varphi^q)$, the equilibrium conditions become

$$\bar{\sigma}.\delta\bar{\varepsilon} = \bar{\sigma}_i D_{i,k}\delta u_k = f_k\delta u_k = f.\delta u, \tag{3}$$

from which follows

$$f_k = \bar{\sigma}_i D_{i,k},$$
$$F^p = (-\bar{\sigma}_1, -\bar{\sigma}_5 + \bar{\sigma}_6, \bar{\sigma}_3 - \bar{\sigma}_4), \qquad M^p = (-\bar{\sigma}_2 l_0, -\bar{\sigma}_3 l_0, -\bar{\sigma}_5 l_0),$$
$$F^q = (\bar{\sigma}_1, \bar{\sigma}_5 - \bar{\sigma}_6, -\bar{\sigma}_3 + \bar{\sigma}_4), \qquad M^q = (\bar{\sigma}_2 l_0, \bar{\sigma}_4 l_0, \bar{\sigma}_6 l_0). \tag{4}$$

From this result we see that $\bar{\sigma}_1$ is the normal force, $\bar{\sigma}_2 l_0$ is the torsion moment, and $\bar{\sigma}_3 l_0$, $\bar{\sigma}_4 l_0$, $\bar{\sigma}_5 l_0$ and $\bar{\sigma}_6 l_0$ are the bending moments at the nodes p and q.

If beams are divided in sufficiently many elements such that for each element the strains remain small, the usual static linear relations between generalized stresses and strains remain valid and can be stated as

$$\bar{\sigma} = S\bar{\varepsilon}, \tag{5}$$
$$S = \mathrm{diag}\{S_1, S_2, S_3, S_4\},$$
$$S_1 = EA/l_0, \quad S_2 = S_t/l_0^3,$$

$$S_3 = \frac{EI_y}{(1+\Phi_z)l_0^3}\begin{pmatrix} 4+\Phi_z & -2+\Phi_z \\ -2+\Phi_z & 4+\Phi_z \end{pmatrix}, \qquad \Phi_z = \frac{12EI_y}{GAk_z l_0^2},$$

$$S_4 = \frac{EI_z}{(1+\Phi_y)l_0^3}\begin{pmatrix} 4+\Phi_y & -2+\Phi_y \\ -2+\Phi_y & 4+\Phi_y \end{pmatrix}, \qquad \Phi_y = \frac{12EI_z}{GAk_y l_0^2}, \tag{6}$$

Here, E is the modulus of elasticity (Young's modulus), G is the shear modulus, A is the area of the cross-section, S_t is the torsional stiffness, I_y and I_z are the area moments of inertia of the cross-section with respect to the two central principal axes, and k_y and k_z are shear coefficients according to [10]. It is seen that the shear deformations can be accounted for by a slightly modified stiffness matrix. This direct tying of the shear deformation to the bending by means of the statics of the beam prevents problems of shear locking. This modification of the element stiffness can already be found in [12]. Initial deformations and material damping according to the Kelvin-Voigt model can be treated by a modification of the stress-strain relation (5) as $\bar{\sigma} = S(\bar{\varepsilon} - \bar{\varepsilon}^0) + S_d\dot{\bar{\varepsilon}}$, where $\bar{\varepsilon}^0$ are the initial deformations and $\dot{\bar{\varepsilon}}$ are the deformation rates, and S_d has the same structure as S.

With the element as described up to this point, all geometrically nonlinear elastostatic problems of frameworks can be solved accurately as long as the beams are divided into sufficiently many elements. Dynamics can be treated if inertial terms as described later on in this section are added. However, in dynamics, more than in statics, it is important that one is able to obtain sufficiently accurate solutions with rather large elements, because decreasing the size of the elements increases, besides the number of degrees of freedom, the largest eigenfrequency in the system and reduces the allowed time step in explicit integration methods. Therefore an inclusion of second-order effects in the stiffness properties of the element is valuable.

For the derivation of the properties of a refined element, a local coordinate system is introduced whose origin coincides with node p and whose axes are in the directions of $\mathbf{e}_x^p$, $\mathbf{e}_y^p$ and $\mathbf{e}_z^p$. A point r on the elastic line of the beam is indicated by the material coordinate s, $0 \le s \le l_0$, and its coordinates in the local frame of reference are $(s + u_x, u_y, u_z)$, where u_x, u_y and u_z are the displacements. The orientation $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ at a point on the elastic line is described by modified Euler angles $(\varphi_x, \varphi_y, \varphi_z)$ as

$$\begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi_x & \sin\varphi_x \\ 0 & -\sin\varphi_x & \cos\varphi_x \end{pmatrix}\begin{pmatrix} \cos\varphi_y & 0 & -\sin\varphi_y \\ 0 & 1 & 0 \\ \sin\varphi_y & 0 & \cos\varphi_y \end{pmatrix}\begin{pmatrix} \cos\varphi_z & \sin\varphi_z & 0 \\ -\sin\varphi_z & \cos\varphi_z & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \mathbf{e}_x^p \\ \mathbf{e}_y^p \\ \mathbf{e}_z^p \end{pmatrix} =$$

$$\begin{pmatrix} \cos\varphi_y\cos\varphi_z & \cos\varphi_y\sin\varphi_z & -\sin\varphi_y \\ -\cos\varphi_x\sin\varphi_z+\sin\varphi_x\sin\varphi_y\cos\varphi_z & \cos\varphi_x\cos\varphi_z+\sin\varphi_x\sin\varphi_y\sin\varphi_z & \sin\varphi_x\cos\varphi_y \\ \sin\varphi_x\sin\varphi_z+\cos\varphi_x\sin\varphi_y\cos\varphi_z & -\sin\varphi_x\cos\varphi_z+\cos\varphi_x\sin\varphi_y\sin\varphi_z & \cos\varphi_x\cos\varphi_y \end{pmatrix}\begin{pmatrix} \mathbf{e}_x^p \\ \mathbf{e}_y^p \\ \mathbf{e}_z^p \end{pmatrix} \tag{7}$$

If we denote a derivative with respect to s by a prime, the six deformations of the one-dimensional continuum are [6]

$$\gamma_x = (\mathbf{r}'.\mathbf{r}')^{1/2} - 1, \qquad \gamma_y = \mathbf{e}_y.\mathbf{r}', \qquad \gamma_z = \mathbf{e}_z.\mathbf{r}',$$

$$\kappa_x = \mathbf{e}_z.\mathbf{e}_y' = -\mathbf{e}_z'.\mathbf{e}_y,$$

$$\kappa_y = \mathbf{e}_z'.\mathbf{e}_x = -\mathbf{e}_z.\mathbf{e}_x', \tag{8}$$

$$\kappa_z = -\mathbf{e}_y'.\mathbf{e}_x = \mathbf{e}_y.\mathbf{e}_x'.$$

γ_x is the extension, γ_y and γ_z are the transverse shear deformations, κ_x is the torsion, and κ_y and κ_z are the bending deformations.

As we are interested in second-order effects, only terms up to the second order have to be retained, while higher-order terms can consistently be disregarded. As the extensional and transverse shear stiffness are much larger than the flexural and torsional stiffness of a beam, γ_x, γ_y and γ_z, and also the axial displacement u_x, can be regarded as second-order effects. The equations (7–8) can now be expanded as

$$\begin{pmatrix} \mathbf{e}_x \\ \mathbf{e}_y \\ \mathbf{e}_z \end{pmatrix} = \begin{pmatrix} 1 - \tfrac{1}{2}\varphi_y^2 - \tfrac{1}{2}\varphi_z^2 & \varphi_z & -\varphi_y \\ -\varphi_z + \varphi_x\varphi_y & 1 - \tfrac{1}{2}\varphi_x^2 - \tfrac{1}{2}\varphi_z^2 & \varphi_x \\ \varphi_y + \varphi_x\varphi_z & -\varphi_x + \varphi_y\varphi_z & 1 - \tfrac{1}{2}\varphi_x^2 - \tfrac{1}{2}\varphi_y^2 \end{pmatrix} \begin{pmatrix} \mathbf{e}_x^p \\ \mathbf{e}_y^p \\ \mathbf{e}_z^p \end{pmatrix}, \tag{9}$$

$$\gamma_x = u_x' + \tfrac{1}{2}u_y'^2 + \tfrac{1}{2}u_z'^2, \qquad \gamma_y = -\varphi_z + u_y', \qquad \gamma_z = \varphi_y + u_z',$$

$$\kappa_x = \varphi_x' - \varphi_y\varphi_z', \qquad \kappa_y = \varphi_y' + \varphi_x\varphi_z', \qquad \kappa_z = \varphi_z' - \varphi_x\varphi_y'. \tag{10}$$

If we introduce the linearized strains $\boldsymbol{\varepsilon}^l$ as

$$\varepsilon_1^l = u_x^q, \ \varepsilon_2^l = l_0\varphi_x^q, \ \varepsilon_3^l = -u_z^q, \ \varepsilon_4^l = l_0\varphi_y^q + u_z^q, \ \varepsilon_5^l = u_y^q, \ \varepsilon_6^l = l_0\varphi_z^q - u_y^q, \tag{11}$$

the non-linear strains up to terms of the second order can be expressed as

$$\bar{\varepsilon}_1 = \varepsilon_1^l + [(\varepsilon_3^l)^2 + (\varepsilon_5^l)^2]/(2l_0), \quad \bar{\varepsilon}_2 = \varepsilon_2^l - (\varepsilon_3^l + \varepsilon_4^l)(\varepsilon_5^l + \varepsilon_6^l)/(2l_0),$$

$$\bar{\varepsilon}_3 = \varepsilon_3^l, \quad \bar{\varepsilon}_4 = \varepsilon_4^l + \varepsilon_2^l\varepsilon_6^l/l_0, \quad \bar{\varepsilon}_5 = \varepsilon_5^l, \quad \bar{\varepsilon}_6 = \varepsilon_6^l - \varepsilon_2^l\varepsilon_4^l/l_0. \tag{12}$$

The displacements and rotations are interpolated in the same way as in the linear case as ($\xi = s/l_0$)

$$u_x = \xi\varepsilon_1^l,$$

$$u_y = (2\xi^2 - \xi^3)\varepsilon_5^l + (-\xi^2 + \xi^3)\varepsilon_6^l + [\Phi_y/(2+2\Phi_y)](\xi - 3\xi^2 + 2\xi^3)(\varepsilon_5^l - \varepsilon_6^l),$$

$$u_z = (-2\xi^2 + \xi^3)\varepsilon_3^l + (\xi^2 - \xi^3)\varepsilon_4^l + [\Phi_z/(2+2\Phi_z)](\xi - 3\xi^2 + 2\xi^3)(-\varepsilon_3^l + \varepsilon_4^l),$$

$$l_0\varphi_x = \xi\varepsilon_2^l,$$

$$l_0\varphi_y = (4\xi - 3\xi^2)\varepsilon_3^l + (-2\xi + 3\xi^2)\varepsilon_4^l + [\Phi_z/(1+\Phi_z)](3\xi - 3\xi^2)(-\varepsilon_3^l + \varepsilon_4^l),$$

$$\epsilon_0\varphi_z = (4\xi - 3\xi^2)\varepsilon_5^1 + (-2\xi + 3\xi^2)\varepsilon_6^1 + [\Phi_y/(1+\Phi_y)](-3\xi + 3\xi^2)(\varepsilon_5^1 - \varepsilon_6^1). \tag{13}$$

The strain energy W^e can be calculated from

$$W^e = \tfrac{1}{2}l_0\int_0^1 [EA\gamma_x^2 + GAk_y\gamma_y^2 + GAk_z\gamma_z^2 + S_t\kappa_x^2 + EI_y\kappa_y^2 + EI_z\kappa_z^2]d\xi. \tag{14}$$

Substituting (10) with the interpolations (13) in the integral (14) yields, if only terms up to the third order are retained,

$$\begin{aligned}
W^e = {} & \tfrac{1}{2}(EA/l_0)(\varepsilon_1^1)^2 + \tfrac{1}{2}(S_t/l_0^2)(\varepsilon_2^1)^2 + \\
& \tfrac{1}{2}(EI_y/l_0^3)\{4(\varepsilon_3^1)^2 - 4\varepsilon_3^1\varepsilon_4^1 + 4(\varepsilon_4^1)^2 - [3\Phi_z/(1+\Phi_z)](\varepsilon_3^1 - \varepsilon_4^1)^2\} + \\
& \tfrac{1}{2}(EI_z/l_0^3)\{4(\varepsilon_5^1)^2 - 4\varepsilon_5^1\varepsilon_6^1 + 4(\varepsilon_6^1)^2 - [3\Phi_y/(1+\Phi_y)](\varepsilon_5^1 - \varepsilon_6^1)^2\} + \\
& [EA/(30l_0^2)]\varepsilon_1^1[17(\varepsilon_3^1)^2 + \varepsilon_3^1\varepsilon_4^1 + 2(\varepsilon_4^1)^2 + 17(\varepsilon_5^1)^2 + \varepsilon_5^1\varepsilon_6^1 + 2(\varepsilon_6^1)^2] + \\
& (S_t/l_0^4)\varepsilon_2^1(-\tfrac{1}{2}\varepsilon_3^1\varepsilon_5^1 - \tfrac{3}{2}\varepsilon_3^1\varepsilon_6^1 + \tfrac{1}{2}\varepsilon_4^1\varepsilon_5^1 - \tfrac{1}{2}\varepsilon_4^1\varepsilon_6^1) + \\
& [(EI_y - EI_z)/l_0^4]\varepsilon_2^1(\varepsilon_3^1\varepsilon_5^1 - \varepsilon_3^1\varepsilon_6^1 - \varepsilon_4^1\varepsilon_5^1 + 3\varepsilon_4^1\varepsilon_6^1)
\end{aligned} \tag{15}$$

With the aid of the inversion of (12), the original barred strains $\bar{\varepsilon}$ can be substituted for the linearized strains. Then stress-strain relations are obtained by taking partial derivatives with respect to the strains. These non-linear relations between generalized stresses and strains could be used in calculations, but we prefer to redefine the generalized strains in such a way that the third-order terms are removed from the expression for the strain energy and the linear relation between generalized stresses and strains remains valid. The reason for this transformation is that it gives a clear separation of physical and geometrical non-linearities; the non-linearities in the present formulation clearly stem from the geometry. Furthermore, the treatment of initial deformations remains simple and constraints such as inextensibility can be specified in a simpler way. It can be checked that the required modified generalized strains, which are denoted by $\boldsymbol{\epsilon}$, are

$$\begin{aligned}
\varepsilon_1 &= \bar{\varepsilon}_1 + (2\bar{\varepsilon}_3^2 + \bar{\varepsilon}_3\bar{\varepsilon}_4 + 2\bar{\varepsilon}_4^2 + 2\bar{\varepsilon}_5^2 + \bar{\varepsilon}_5\bar{\varepsilon}_6 + 2\bar{\varepsilon}_6^2)/(30l_0), \\
\varepsilon_2 &= \bar{\varepsilon}_2 + (-\bar{\varepsilon}_3\bar{\varepsilon}_6 + \bar{\varepsilon}_4\bar{\varepsilon}_5)/l_0, \\
\varepsilon_3 &= \bar{\varepsilon}_3 + \bar{\varepsilon}_2(\bar{\varepsilon}_5 + \bar{\varepsilon}_6)/(6l_0), \\
\varepsilon_4 &= \bar{\varepsilon}_4 - \bar{\varepsilon}_2(\bar{\varepsilon}_5 + \bar{\varepsilon}_6)/(6l_0), \\
\varepsilon_5 &= \bar{\varepsilon}_5 - \bar{\varepsilon}_2(\bar{\varepsilon}_3 + \bar{\varepsilon}_4)/(6l_0), \\
\varepsilon_6 &= \bar{\varepsilon}_6 + \bar{\varepsilon}_2(\bar{\varepsilon}_3 + \bar{\varepsilon}_4)/(6l_0).
\end{aligned} \tag{16}$$

The corresponding generalized stresses σ have now a slightly modified meaning in the case of finite deformations

The additional terms in ε_1 take into account the additional elongation of the beam axis caused by its bending and the additional terms in ε_2 measure the extra torsion of the beam caused by its bending. If the flexural rigidity is equal in all directions, the strains ε_3 to ε_6 need not be modified as can be seen from (15), the additional terms take into account the effect of unsymmetrical bending caused by a twist of the beam. In planar problems, $\bar{\varepsilon}_2$, $\bar{\varepsilon}_3$ and $\bar{\varepsilon}_4$ are zero and only ε_1 is modified. Because the rigidity against elongation is much larger than the flexural and torsional rigidity, the modification of ε_1 is the most important one and the other modifications are only relevant to more special cases, for instance when one has large differences in flexural rigidities, the beam is used as a shaft for the transmission of a torque, or one has initial deformations in precurved and pretwisted beams.

The inertia terms are calculated in the same way as in [9]. The interpolation for the elastic line for finite deformations is now taken to be

$$\mathbf{r}(\xi) = (1 - 3\xi^2 + 2\xi^3)\mathbf{x}^p + (\xi - 2\xi^2 + \xi^3)(l_0 + \varepsilon_1)\mathbf{e}_x^p + (3\xi^2 - 2\xi^3)\mathbf{x}^q + (-\xi^2 + \xi^3)(l_0 + \varepsilon_1)\mathbf{e}_x^q . \tag{17}$$

In many cases ε_1 is small and can be neglected; it is only included in (17) in order to obtain the classical mass matrix for small longitudinal vibrations. Even if longitudinal vibrations are important, better approximations of the eigenfrequencies are obtained if this term is neglected. If we put ε_1 equal to zero, and the mass per unit of length is m, we obtain the mass matrix and convective inertial terms by evaluating the integral

$$ml_0\int_0^1 \delta\mathbf{r}.\ddot{\mathbf{r}}\,d\xi. \tag{18}$$

If the rotations of the nodes are parametrized by φ^p and φ^q, this results in the mass matrix

$$\mathbf{M} = \frac{ml_0}{420}\begin{pmatrix} 156\,\mathrm{I} & 22l_0\mathbf{A} & 54\mathrm{I} & -13l_0\mathbf{B} \\ & 4l_0^2\mathbf{A}^T\mathbf{A} & 13l_0\mathbf{A}^T & -3l_0^2\mathbf{A}^T\mathbf{B} \\ & & 156\mathrm{I} & -22l_0\mathbf{B} \\ \text{symm.} & & & 4l_0^2\mathbf{B}^T\mathbf{B} \end{pmatrix}, \tag{19}$$

and the inertia terms which are quadratic in the velocities

$$-\mathbf{f}_{in} = \frac{ml_0}{420}\begin{pmatrix} l_0(22\mathbf{A}'\dot{\varphi}^p\dot{\varphi}^p - 13\mathbf{B}'\dot{\varphi}^q\dot{\varphi}^q) \\ l_0^2(4\mathbf{A}^T\mathbf{A}'\dot{\varphi}^p\dot{\varphi}^p - 3\mathbf{A}^T\mathbf{B}'\dot{\varphi}^q\dot{\varphi}^q) \\ l_0(13\mathbf{A}'\dot{\varphi}^p\dot{\varphi}^p - 22\mathbf{B}'\dot{\varphi}^q\dot{\varphi}^q) \\ l_0^2(-3\mathbf{B}^T\mathbf{A}'\dot{\varphi}^p\dot{\varphi}^p + 4\mathbf{B}^T\mathbf{B}'\dot{\varphi}^q\dot{\varphi}^q) \end{pmatrix} \tag{20}$$

where

$$\mathbf{A} = \partial\mathbf{e}_x^p/\partial\varphi^p, \quad \mathbf{B} = \partial\mathbf{e}_x^q/\partial\varphi^q, \quad \mathbf{A}' = \partial\mathbf{A}/\partial\varphi^p, \quad \mathbf{B}' = \partial\mathbf{B}/\partial\varphi^q \tag{21}$$

The rotary inertia of the cross-section can be lumped onto the nodes, half of the total rotary inertia of the beam on each node.

The equations of motion for a complete system are formed by the algorithm of [8,9], which is based on the principle of virtual power and uses a minimal set of differential equations.

3. Application to Elastic Stability Problems

In order to check whether the geometric stiffness is represented properly in the described element, two spatial linear buckling problems are analysed for which analytical solutions are known. In these examples, the influence of transverse shear is not considered.

The first example is the lateral buckling, or tipping, of a clamped-free beam with a narrow rectangular cross-section which is loaded by a transverse force at its end in the direction of the larger flexural rigidity. The theoretical buckling load is $F_{th} = 4.013(EIS_t)^{1/2}/l^2$, where EI is the smaller flexural rigidity and l is the length of the beam. For the numerical analysis, the beam is divided in one, two or four equal elements and the two cases in which the second order terms in the generalized strains are retained or not are considered. The results are given in Table 1, where F^1_{cr} are the critical loads which are obtained without the second-order terms and F^2_{cr} are the critical loads when these terms are included in the analysis.

Table 1. Critical loads for lateral buckling by a force.

number of elements	F^1_{cr}/F_{th}	F^2_{cr}/F_{th}
1	infinite	1.4951
2	1.2180	1.0690
4	1.0459	1.0153

We observe that with the inclusion of the second-order terms in the bending deformations, the error in the buckling load is decreased by a factor of about three, while the order of convergence remains two. Both approximations converge to the theoretical value.

Table 2. Critical loads for buckling by a torsion moment.

number of elements	M^1_{cr}/M_{th}	M^2_{cr}/M_{th}
1	infinite	1.1027
2	1.2732	1.0075
4	1.0548	1.0005

The second example is the buckling of a clamped-free circular shaft by a torsion moment at its free end If the torsion moment is semitangential, the theoretical buckling load is $M_{th} = \pi EI/l$ [13]. Table 2 shows the results of the calculations, where the symbols have a similar meaning as in Table 1. We observe that the inclusion of second-order terms raises the order of convergence from two to four in this case.

With the inclusion of the second-order terms in the strains, we obtain, apart from numerical errors, the same answers as Argyris *et al.* [14]. This is also the case in the other examples of [14], which are not reported on here.

The order of convergence in the lateral buckling problem remains two, because the buckling mode involves torsion, which is approximated by a linearly varying torsion angle If a polynomial approximation of the third degree is used, a higher order of convergence can be obtained [15].

4. Slider-Crank Mechanism with a Flexible Connecting Rod

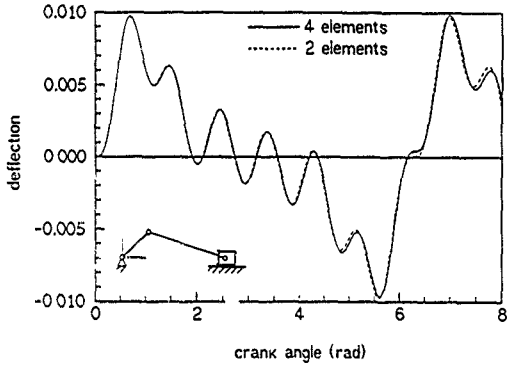


Figure 1. Dimensionless deflection of the centre point of the connecting rod of a slider-crank mechanism. $\omega = 124.8$ rad/s, $m_{block} = 0$.

The transient response of a slider-crank mechanism with a rigid crank and a flexible connecting rod has been the subject of a considerable number of studies Chu and Pan [16] treated among other cases the example in which the crank has a length of 0.1524 m, the connecting rod has a length of 0.3048 m and a circular cross-section with a diameter of 0.00635 m and the slider block is massless The material of the connecting rod is steel with a density of $\rho = 7834\ 56$ kg/m^3, a Young's modulus $E = 2\ 06843.10^{11}$ N/m^3, a Poisson's ratio $\nu = 0\ 285$ and no material damping The constant rotation speed of the crank is $\omega =$

124.8 rad/s The motion is simulated over a rotation angle of the crank of 8 rad, starting from the initial conditions that the deformation and deformation rate of the connecting rod are zero. The measured output quantity is the lateral deflection of the centre point of the connecting rod divided by its length.

As a reference we modelled the connecting rod with four planar beam elements, with all effects included (12 degrees of freedom). The results are given as the solid line in Fig. 1. If the connecting rod is modelled by two beam elements and the normal and shear deformation as well as the rotary inertia are neglected (4 degrees of freedom), the results given by the dashed line are obtained. As the system has no damping, any small change in modelling or numerical integration method will give large differences of solutions at large times. Therefore a good measure for the comparison of results seems to be the value of the first maximum of the deflection. These values are listed in Table 3 and are compared with some results from the literature

Table 3. Comparison of the results of different calculations of the deflections of the connecting rod of a slider-crank mechanism

reference	number of elements or modes	value of the first maximum
present	4 elements	0.00976
present	2 elements	0.00973
[16]	1 mode	0.0105
[17]	3 elements	0.0094
[18]	2 elements	0.0102
[19]	3 elements	0.0098
[20]	2 elements	0.0097

We remark that the results of [16] seem to overestimate the first maximum and the geometric non-linearity was not accounted for in [18], [19] and [20] give results quite similar to those in Fig. 1.

As the rotation speed in this example is rather low and the influence of the coupling of the gross motion and the deformation is small, the same mechanism is studied in the case that the rotation speed of the crank is 150 rad/s and the mass of the slider block is equal to half the mass of the connecting rod. The results of simulations with two and four elements are shown in Fig. 2. The values of the first maximum are now 0.01394 (four elements) and 0.01391 (two elements). Results from studies in which the geometric non-linearity is not included [21, 22, 23] overestimate this value, on the other hand, a study in which this effect is included [24], underestimates the maximum.

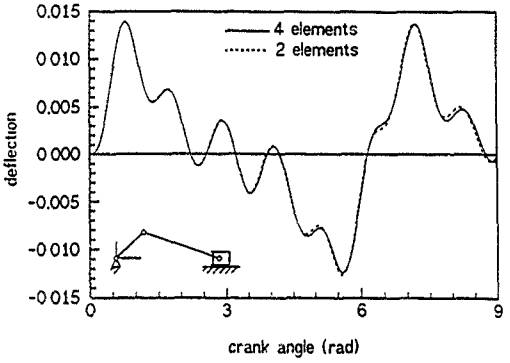


Figure 2. Dimensionless deflection of the centre point of the connecting rod of a slider-crank mechanism. $\omega = 150$ rad/s, $m_{block} = 0.0378126$ kg.

b. Spin-up Motion of a Beam

5.1. PLANAR SYSTEM

The next example system consists of a beam whose base is attached to a shaft orthogonal to the beam axis, which is given a prescribed rotation. In particular the spin-up motion is considered, in which the beam is accelerated from rest to a constant angular velocity. The prescribed angle of rotation ψ is given by

$$\psi = \frac{\omega^*}{T^*}\{\tfrac{1}{2}t^2 + \frac{T^{*2}}{4\pi^2}(\cos\frac{2\pi t}{T^*} - 1)\}, \qquad 0 \le t \le T^*,$$

$$\psi = \omega^*(t - \tfrac{1}{2}T^*), \qquad t > T^*. \tag{22}$$

Firstly we take the case from [25, 26], in which the length of the beam is $L = 8$ m, the beam has a rectangular cross-section of 0 03675 m by 0.001986 m and is made of aluminium with $\rho = 2766\,67$ kg/m^3, $E = 6\,895.10^{10}$ N/m^2, and $\nu = 0.32$ The final angular velocity is $\omega^* = 4$ rad/s, and the spin-up time is $T^* = 15$ s The beam is modelled by four and two elements, as before, and the lateral deflection of the tip of the beam is taken as an observed quantity, for which results are given in Fig. 3 The extreme value of the deflection is −0 5360 m (four elements) and −0 5338 m (two elements)

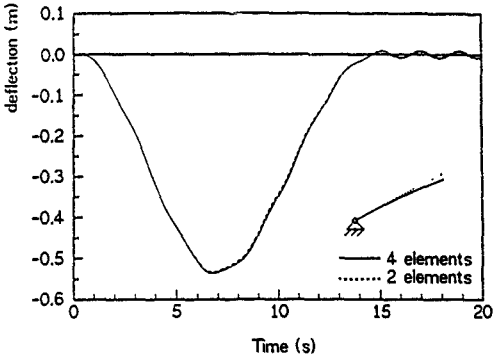


Figure 3. Lateral deflection for the spin-up motion of [25, 26].

The extreme value in [25, 26] is −0.543 m in their most accurate model; the general shape and the residual vibrations for $t > 15$ s agree with our results. Apparently, the results given in [27] contain large errors.

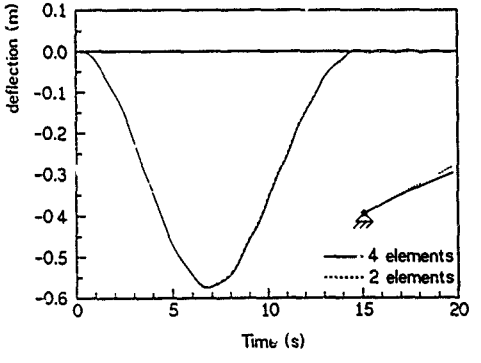


Figure 4. Lateral deflection of the spin-up motion of [28].

Secondly, we take the case of [28], with $L = 10$ m, $\rho A = 1\,2$ kg/m, $EI = 14000$ Nm^2, $\rho I = 0.0006$ kgm, $GAk = 1\,10^7$ N, $EA = 2.8.10^7$ N, and $\omega^* = 6$ rad/s, $T^* = 15$ s The results for this case are shown in Fig. 4. The extreme values of the deflection are now −0 5739 m (four elements) and −0 5726 m (two elements); the amplitude of the vibrations for $t > 15$ s is 0.0037 m. If we compare these results with those from the literature, we observe that the extreme deflection is about 1% lower in [28], about 1% higher in [29] and about 2% higher in [30]. The amplitude of the residual vibrations in [29, 30] is higher.

In general we may conclude that several calculations give comparable results for this problem. With the beam element presented in this paper we were able to obtain an error of less than 1% with a model with two elements having four degrees of freedom, while others needed more elaborate models to obtain the same accuracy.

5.2. SPATIAL SYSTEM

In [1], the spin-up motion was considered for the case in which part of the beam had an open channel cross-section. The beam element we have described here is not intended to solve problems involving beams with an open thin-walled cross-section. The torsional stiffness is very low, which causes that effects which may safely be neglected for beams with a solid cross-section can be important, such as the constraint of warping at the ends, the eccentricity of the centre of shear, the shortening of beam due to torsion, and the distortion of the shape of the cross-section. Furthermore, local phenomena and the distribution of residual stresses may become important. It is advisable to develop special elements for problems involving beams of this nature, or to use plate and shell elements.

Some aspects of thin-walled open cross-section beams can be included approximately into the analysis with the present element. The effect of the constrained warping can be approximated by introducing an effective torsional stiffness which gives the same relative rotation angle of the end nodes if the beam is loaded by a torque. This is a good approximation if the influence of the warping constraint is confined within zones near the ends of the beam, in which case one may introduce the effective length of the beam, $L_e = L - (EI/S_t)^{1/2}$, where I is the warping constant.

The effect of the eccentricity of the shear centre can be accommodated for by modelling the beam with two parallel elements, one along the line of centroids, and one along the line of shear centres, whose ends are rigidly eccentrically connected. The mass and normal stiffness are attributed to the centroid element, and the torsional and flexural rigidity are attributed to the shear centre element.

The beam considered in [1] has a length of 16 m and is built up from a rigid beam of length 8 m, a doubly symmetric tube of length 2½ m, and a channel of length 5½ m. The material is aluminium with $\rho = 2766.67$ kg/m^3, $E = 6\,895.10^{10}$ N/m^2, and $G = 2\,6519.10^{10}$ N/m^2. The properties of the tube are $A = 0\,000384$ m^2, $I = 1.5.10^{-7}$ m^4, and $S_t = 5834.18$ Nm^2, and those of the channel are $A = 7.3.10^{-5}$ m^2, $I_y = 4\,8746.10^{-9}$ m^4, $I_z = 8.2181.10^{-9}$ m^4, $S_t = 0\,645207$ Nm^2, $I = 5.0156.10^{-13}$ m^6, and the eccentricity of the shear centre is $z_0 = -0.01875$ m. Initially the beam is located along the global X-axis, while the shaft is in the direction of the global Z-axis. In the modelling of this

example, one beam element is used for the tube and four equal pairs of beam elements for the channel. Effects of shear deformation and normal force deformation are not included. The spin-up time is $T^* = 15$ s, and the final angular velocity $\omega^* = 6$ rad/s.

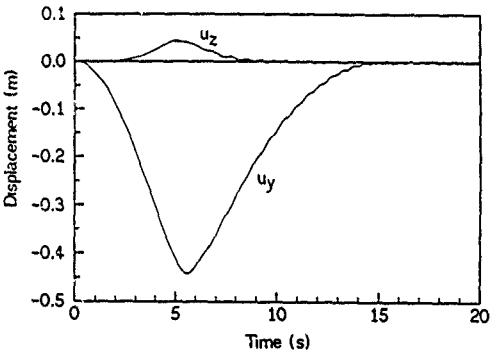


Figure 5. Deflection of the tip.

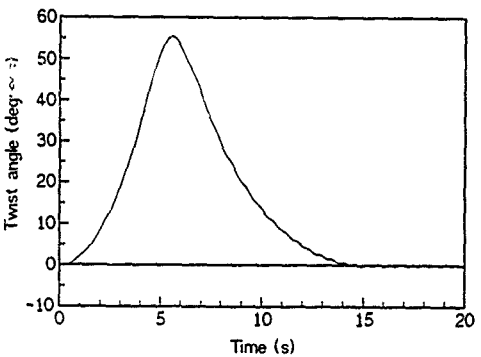


Figure 6 Total torsion angle at the tip

Figure 5 shows the deflection of the centroid of the tip in the two directions perpendicular to the undeformed axis of the beam. Figure 6 shows the total torsion angle of the beam. The most conspicuous difference from the results of [1] is that u_z has the opposite sign and returns to zero when the beam rotates at a constant angular velocity. This difference is due to the fact that the theory in [1] is based on the assumption of small deflections and an error in the expression for the strain energy Furthermore, the deflections shown here are larger than those in [1]. This is also caused by a softening effect in the torsional rigidity for large torsion angles in our present element.

5 Concluding Remarks

The present paper has pointed out the importance of the inclusion of the geometric non-linearity in the modelling of beams and the validation of results obtained with these elements. The examples treated here, however, can only be considered as a commencement of the process of validation, especially as few spatial examples were considered. Therefore, a further study seems appropriate.

For slender beams, the effects of transverse shear deformation and elongation of the beam axis appear to be small. The rotary inertia of the cross-section is only important if torsional vibrations can occur.

From the relative ease with which it was possible to include second-order terms in the description of the stiffness of the beam element, one might expect that an extension to higher orders of approximation is straightforward. However, a consistent inclusion of all terms is involved and even seems to be unnecessary for dynamical problems, because some minimal number of degrees of freedom have to be retained.

References

[1] T.R. Kane, R.R. Ryan and A.K. Banerjee, Dynamics of a cantilever beam attached to a moving base, *AIAA Journal of Guidance, Control, and Dynamics* 10(2) (1987), 139–151.

[2] W. Visser and J.F. Besseling, Large displacement analysis of beams, Report WTHD-10, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1969.

[3] J.F. Besseling, Non-linear analysis of structures by the finite element method as a supplement to a linear analysis, *Computer Methods in Applied Mechanics and Engineering* 3 (1974), 173–194.

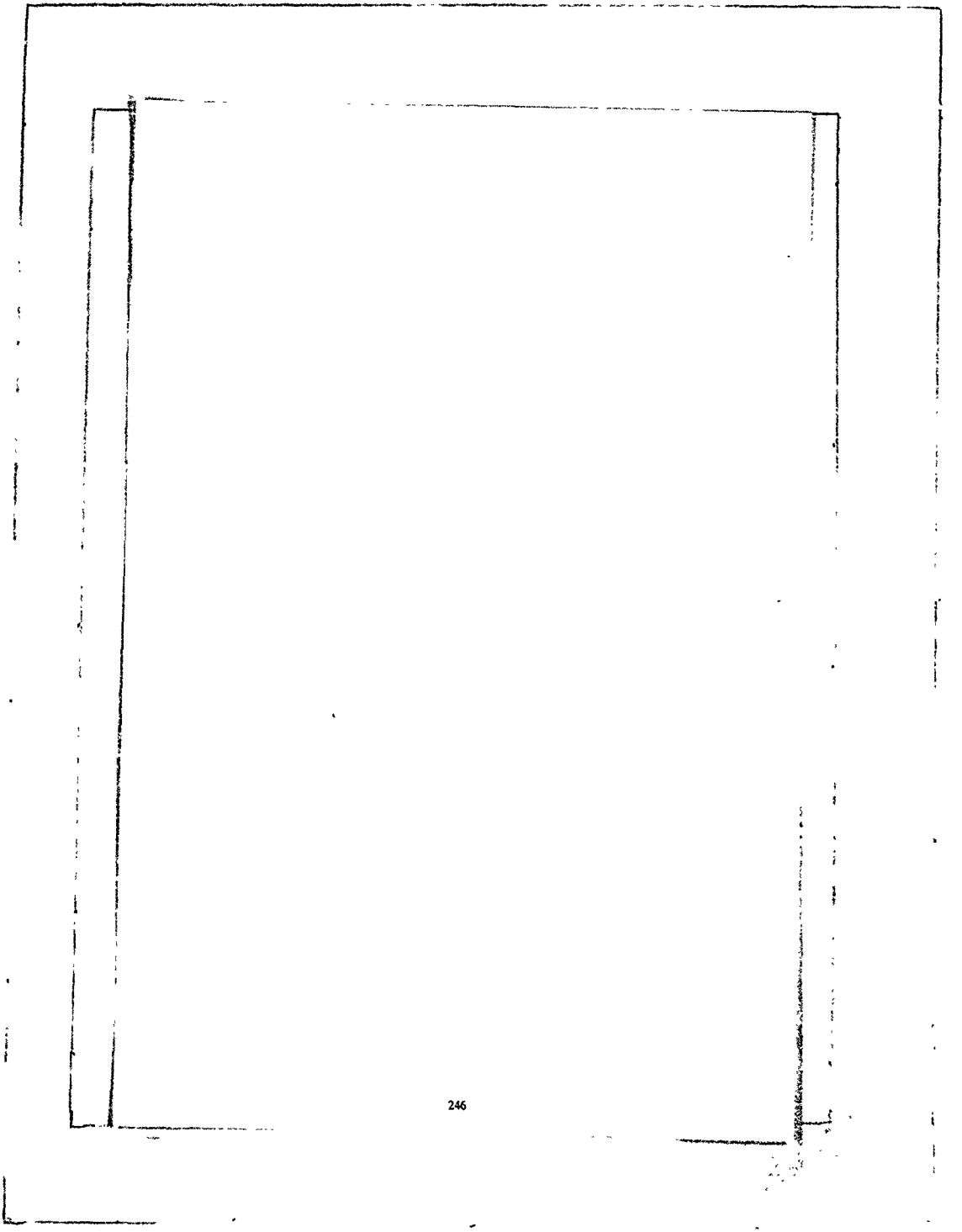
[4] J.F Besseling, Post-buckling and non-linear analysis by the finite element method as a supplement to a linear analysis, *Zeitschrift für angewandte Mathematik und Mechanik* 55 (1975), T3–T15.

[5] J.F. Besseling, Derivatives of deformation parameters for bar elements and their use in buckling and postbuckling analysis, *Computer Methods in Applied Mechanics and Engineering* 12 (1977), 97–124.

[6] J.F. Besseling, Non-linear theory for elastic beams and rods and its finite element representation, *Computer Methods in Applied Mechanics and Engineering* 31 (1982), 205–220.

[7] K van der Werff and J B Jonker, Dynamics of flexible mechanisms. In E.J. Haug (ed), *Computer Aided Analysis and Optimization of Mechanical System Dynamics*. Springer, Berlin, 1984, 381–400.
[8] J.B. Jonker, A finite element dynamic analysis of spatial mechanisms with flexible links, *Computer Methods in Applied Mechanics and Engineering* 76 (1989), 17–40.
[9] J.P. Meijaard, Direct determination of periodic solutions of the dynamical equations of flexible mechanisms and manipulators, *International Journal for Numerical Methods in Engineering* 32 (1991), 1691–1710.
[10] G.R. Cowper, The shear coefficient in Timoshenko's beam theory, *ASME Journal of Applied Mechanics* 33 (1966), 335–340.
[11] J.H. Argyris, Continua and discontinua, an aperçu of recent developments of the matrix displacement methods. In: J.S. Przemieniecki, R.M. Bader, W.F Bozich, J R. Johnson and W.J. Mykytow (eds), *Matrix Methods in Structural Mechanics*, Wright–Patterson Air Force Base, Dayton, Ohio, 1966, 11–189.
[12] J.S. Przemieniecki, *Theory of Matrix Structural Analysis*, McGraw–Hill, New York, 1968. Reprinted by Dover, New York, 1985.
[13] H. Ziegler, *Principles of Structural Stability*, Blaisdell, Waltham MA, 1968.
[14] J.H. Argyris, P.C. Dunne, G. Malejannakis and D.W. Scharpf, On large displacement–small strain analysis of structures with rotational degrees of freedom, *Computer Methods in Applied Mechanics and Engineering* 14 (1978), 401–451; 15 (1978), 99–135.
[15] R.S. Barsoum and R.H. Gallagher, Finite element analysis of torsional and torsional–flexural stability problems, *International Journal for Numerical Methods in Engineering* 2 (1970), 335–352.
[16] S.–C. Chu and K.C. Pan, Dynamic response of a high–speed slider–crank mechanism with an elastic connecting rod, *ASME Journal of Engineering for Industry* B97 (1975), 542–550.
[17] J.O. Song and E.J. Haug, Dynamic analysis of planar flexible mechanisms, *Computer Methods in Applied Mechanics and Engineering* 24 (1980), 359–381.
[18] A Shabana and R.A. Wehage, Variable degree–of–freedom component mode analysis of inertia variant flexible mechanical systems, *ASME Journal of Mechanisms, Transmissions, and Automation in Design* 105 (1983), 371–378
[19] W.P. Koppens, A.A.H.J. Sauren, F.E. Veldpaus and D.H. van Campen, The dynamics of a deformable body experiencing large displacements, *ASME Journal of Applied Mechanics* 55 (1988), 676–680.
[20] K.–M. Hsiao and J.–Y. Jang, Dynamic analysis of planar flexible mechanisms by co–rotational formulation, *Computer Methods in Applied Mechanics and Engineering* 87 (1991), 1–14.
[21] O.P. Agrawal and A.A. Shabana, Dynamic analysis of multibody systems using component modes, *Computers & Structures* 21 (1985), 1303–1312.
[22] O P. Agrawal and A.A. Shabana, Application of deformable–body mean axis to flexible multibody system dynamics, *Computer Methods in Applied Mechanics and Engineering* 56 (1986), 217–245.
[23] D.F. Bartolone and A.A. Shabana, Effect of beam initial curvature on the dynamics of deformable multibody systems, *Mechanism and Machine Theory* 24 (1989), 411–429.
[24] E.M. Bakr and A.A. Shabana, Geometrically nonlinear analysis of multibody systems, *Computers & Structures* 23 (1986), 739–751.

[25] S.-S. Kim and E.J Haug, A recursive formulation for flexible multibody dynamics, Part I open-loop systems, *Computer Methods in Applied Mechanics and Engineering* 71 (1988), 293–314.
[26] S.-C. Wu and E J. Haug, Geometric non-linear substructuring for dynamics of flexible mechanical systems, *International Journal for Numerical Methods in Engineering* 26 (1988), 2211–2226.
[27] Z.E. Boutaghou, A G Erdman and H.K. Stolarski, Dynamics of flexible beams and plates in large overall motions, *ASME Journal of Applied Mechanics* 59 (1992), 991–999.
[28] J.C. Simo and L. Vu-Quoc, On the dynamics in space of rods undergoing large motions – a geometrically exact approach, *Computer Methods in Applied Mechanics and Engineering* 66 (1988), 125–161.
[29] S.K. Ider and F.M.L. Amirouche, Nonlinear modeling of flexible multibody systems dynamics subjected to variable constraints, *ASME Journal of Applied Mechanics* 56 (1989), 444–450.
[30] J.D. Downer, K.C Park and J.C. Chiou, Dynamics of flexible beams for multibody systems: a computational procedure, *Computer Methods in Applied Mechanics and Engineering* 96 (1992), 373–408.



ELIMINATION OF CONSTRAINT EQUATIONS FROM FLEXIBLE MECHANICAL SYSTEMS

D. M. Russell and K. D. Willmert
Clarkson University
Mechanical and Aeronautical Engineering Department,
Potsdam, NY 13699, USA.

ABSTRACT. A method for modeling planar and spatial mechanisms is presented which is capable of efficiently predicting the nonlinear transient response of flexible systems Large deformations are assumed and nonlinear geometric properties are encorporated. In this research, each element of a mechanism is treated as an individual body, with its own flexible and rigid body modes. Constraints are used to enforce compatibility conditions between the individual *bodies* which make up a mechanism. One of the unique features of this study is how the constraint equations are handled. They are initially introduced into the global system of equations using Lagrange multipliers. The classical solution technique is to numerically solve for the Lagrange multipliers and the generalized coordinates together. In this study the Lagrange multipliers are first analytically eliminated from the equations of motion. As a result the constraint equations are encorporated directly into the system of ordinary differential equations of motion, without the need to solve for the Largange multipliers.

1 Introduction

Most of the work in mechanism deformation analysis is based on the linear theory of elasticity [1–5]. In most of these, a kinematics and rigid body analysis is used to solve for the gross body motion and the inertial forces. By employing different techniques, the elastic deformations are then found by applying these inertial forces as externally applied forces to the linear elasticity problem. The small elastic deformations are superimposed onto the gross motion in order to predict the total motion of each link in the system.

Recently, several researchers have presented methods which couple the gross body motion and the elastic deformations [4–9]. In these methods the configuration of each elastic body is identified using two sets of coordinates. These generalized sets of coordinates define the location and orientation of a reference frame, with respect to an inertial frame, and the elastic coordinates that describe the deformation, defined with respect to the body reference. The equations of motion for the mechanism are then expressed in terms of these two sets of coordinates. Depending on the required type of motion, operating speed and restrictions on the design, some mechanisms will exhibit large deformations. These large deformations necessitate a nonlinear dynamic deformation and stress analysis. Some research based on nonlinear analysis can be found in references [6–12].

The purpose of this study is to develop modeling and solution techniques, and subsequently, a computer program capable of efficiently analyzing the nonlinear transient response of planar and spatial mechanisms. In this research, each element of a mechanism

is treated as an individual body, with its own flexible and rigid body modes. Constraint equations are then used to enforce the necessary compatibility conditions between the individual *bodies* which make up the mechanism. By doing this, a mechanism can be modeled as a multibody dynamic system.

Many methods have been used to represent the holonomic and nonholonomic constraints of rigid [13–16] and flexible [17, 18] multibody systems. The classical approach for handling constraints, using Lagrange multipliers, has the drawback of increasing the number of unknowns by the number of constraints. Other methods for handling constraints include the use of; a penalty formulation [19], orthogonal compliment arrays [20], recursive formulations [21], and component mode synthesis [22]. Most of the recent work in constrained multibody dynamics use the classical Lagrange multiplier approach either in conjunction with other approaches, such using bond graphs [23], or various mathematical operations, to reduce the number of degrees of freedom and/or increase the stability of the problem [24]. One of the unique features of this study is how the constraint equations are handled. In this study the constraints are introduced into the system of equations using Lagrange multipliers. The Lagrange multipliers are then solved for directly and substituted back into the system of equations. The result is a global system of equations which encorporates the constraints without increasing the degrees of freedom of the system.

2 Generalized Beam Element Kinematics

The displacements, velocities and accelerations of a general flexible beam element which is part of a mechanism, undergoing rigid-body and flexible deformations, is presented in this section with respect to a fixed global coordinate system, G.

The rigid body orientation of a beam element is described by a rigid body translation and a sequence of rotations. The local coordinate systems, H and E, of the beam element are attached to the endpoint of that element. Coordinate system H differs from the inertial coodinate system, G, by a translation of $\vec{X}_G^T = [\, X_G \; Y_G \; Z_G \,]$ and coordinate system **E** is defined to have the same origin as H with its *x-axis*, $\hat{\mathbf{e}}_{xE}$, colinear with the elastic axis of the beam element. The position vector, relative to local coordinate system, E, of any material point P, on an elastic beam element can be described as,

$$\begin{aligned} \vec{r}_{PD} &= (x_p + u)\hat{\mathbf{e}}_{xE} + (v)\hat{\mathbf{e}}_{yE} + (w)\hat{\mathbf{e}}_{zE} + y_p\hat{\mathbf{e}}_{yD} + z_p\hat{\mathbf{e}}_{zD} \\ &= \left(\vec{x}_H^T + \vec{U}^T\right)\hat{\mathbf{e}}_E + \vec{x}_p^T\hat{\mathbf{e}}_D \end{aligned} \tag{1}$$

where, $\vec{x}_H^T = [\, x_p \; 0 \; 0 \,]$ is the distance along the elastic axis, $\vec{x}_p^T = [\, 0 \; y_p \; z_p \,]$ is the position in the cross section of the beam, $\vec{U}^T = [\, u \; v \; w \,]$ are the elastic deformations at x_p, and $\hat{\mathbf{e}}_D$ is the deformed element's coordinate system, relative to E. The position vector of this point, at any time, relative to the inertial coordinate system, G, is illustrated in Figure 1, and can be expressed as,

$$\vec{R}_{PD} = \vec{R}_{GH} + \vec{r}_{PD} \tag{2}$$

where,

$$\vec{R}_{GH} = (X_G)\hat{\mathbf{e}}_{xG} + (Y_G)\hat{\mathbf{e}}_{yG} + (Z_G)\hat{\mathbf{e}}_{zG} = \left(\vec{X}_G^T\right)\hat{\mathbf{e}}_G \tag{3}$$

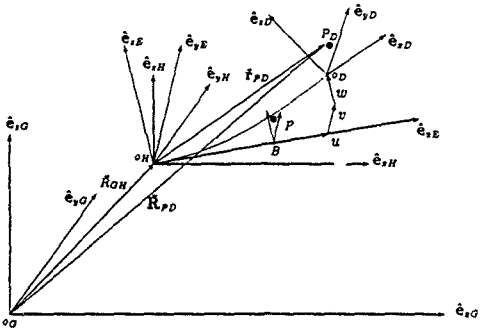


Figure 1: Position Vector of Point P_D on the Deformed Element.

Equation (2) is defined in terms of the inertial, local and deformed systems of coordinates, G, E, and D respectively. Transformation matrices are needed to express this equation strictly in terms of the global coordinate system The rigid-body transformation matrix is,

$$\mathbf{T}_{GE} = \begin{bmatrix} C_y C_z & C_y S_z & -S_y \\ S_x S_y C_z - C_x S_z & S_x S_y S_z + C_x C_z & S_x C_y \\ C_x S_y C_z + S_x S_z & C_x S_y S_z - S_x C_z & C_x C_y \end{bmatrix} \tag{4}$$

By using a second order therory of elasticity the transformation between the undeformed and deformed coordinate systems can be defined as,

$$\mathbf{T}_{ED} = \begin{bmatrix} 1 & v_{,x} & w_{,x} \\ -(w_{,x}\theta + v_{,x}) & 1 & \theta \\ -(w_{,x} - v_{,x}\theta) & -(\theta + v_{,x}w_{,x}) & 1 \end{bmatrix} \tag{5}$$

The transformations of the coordinate systems are then defined by, $\hat{\mathbf{e}}_G = \mathbf{T}_{GE}\hat{\mathbf{e}}_H$ and, $\hat{\mathbf{e}}_G = \mathbf{T}_{GE}\mathbf{T}_{ED}\hat{\mathbf{e}}_D$. By substituting these relations into Equation (2), $\vec{\mathbf{R}}_{PD}$ becomes,

$$\vec{\mathbf{R}}_{PD} = \left[\vec{\mathbf{X}}_G + \left(\vec{\mathbf{x}}_H + \mathbf{T}_{GE}\vec{\mathbf{U}}\right) + \mathbf{T}_{GE}\mathbf{T}_{ED}(\vec{\mathbf{x}}_p)\right]^T \hat{\mathbf{e}}_G \tag{6}$$

$\vec{\mathbf{R}}_{PD}$ now represents the position vector of a general point, P_D, on the element, in terms of the fixed system of coordinates G. A complete derivation of this can be found in reference [25].

The elastic deformations of the beam element are obtained using a Finite Element Method First order polynomials are chosen for the shape functions of the axial displacement, u, and the twist about the x_H-axis, θ, of the element These values are expected to be relatively small when compared to the the v and w directions. The transverse deformations, v and w, are described using fifth degree Hermite polynomials in order to ensure continuous curvature between elements [1,10]

3 Elemental Equations of Motion

In this study a variable correlation table and a set of constraint equations are used to assemble the elemental equations into the global system of equations of motion for a complex flexible multibody system. The variable correlation table is used to identify the rigid body and flexible degrees of freedom of the mechanism and the constraint equations relate the degrees of freedom of one element to the degrees of freedom of another element. These constraints are introduced into the system of equations using Lagrange multipliers. The Lagrange multipliers are then solved for directly and substituted back into the system of equations. This practice and the use of a variable correlation table avoids increasing the degrees of freedom of the system and allows for a wide variety of methodologies, ranging from the classical Finite Element approach to a complete flexible multibody dynamics analysis, where each element is considered to be a separate entity.

Lagrange's equation is applied to a general flexible beam element in the following section. This includes the kinetic energy, strain energy and externally applied forces. The variable correlation table and the constraint functions are then presented, followed by the global system of equations of motion for a general multibody system. This system of equations maintains a complete coupling between the rigid body and flexible modes, and is used to solve for the rigid body motion and flexible deformations simultaneously.

3.1 LAGRANGE'S EQUATIONS

The matrix differential equations of motion for a single finite element of a mechanism is developed by employing Lagrange's equations. This system of elemental equations of motions can then be assembled into a global set of equations for the entire mechanism.

The Lagrange's equations for an element can be represented as,

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\bar{\mathbf{Q}}}_e}\right) - \frac{\partial \mathcal{L}}{\partial \bar{\mathbf{Q}}_e} = 0 \tag{7}$$

where, the vector $\bar{\mathbf{Q}}_e$ contains the complete set of generalized coordinates for the general beam element, and $\mathcal{L}$ is the Lagrangian of the element, which is defined as,

$$\mathcal{L} = KE - SE - F_e \tag{8}$$

where, KE, SE and F_e are the kinetic energy, strain energy and externally applied forces of an element, respectively. By extracting the generalized velocities, the kinetic energy can be expressed as,

$$KE = \frac{1}{2}\dot{\bar{\mathbf{Q}}}_e^T [\mathbf{M}_e] \dot{\bar{\mathbf{Q}}}_e \tag{9}$$

where the generalized velocities, $\dot{\bar{\mathbf{Q}}}_e$, are defined by,

$$\dot{\bar{\mathbf{Q}}}_e^T = \left[\dot{\bar{\mathbf{X}}}_G^T \;\; \dot{\bar{\theta}}_G^T \;\; \dot{\bar{\mathbf{q}}}_e^T \right] \tag{10}$$

where, $\dot{\bar{\mathbf{X}}}_G^T$ are the rigid body displacments, $\dot{\bar{\theta}}_G^T$ are the rigid body rotations, and $\dot{\bar{\mathbf{q}}}_e^T$ are the flexible deformations. Similarly the strain energy can be expressed as,

$$SE = \frac{1}{2}\bar{\mathbf{Q}}^T \mathbf{K}_e \bar{\mathbf{Q}} \tag{11}$$

The modeling of the gravitational force is straight forward and is characterized by a gravitational vector. By defining $\vec{G} = g\hat{h}_g$, where g is the gravitational constant and $\hat{h}_g$ is the unit vector that defines the direction of action of the gravitational force, the force on an element is,

$$F_g = \vec{R}_{PD}^T \vec{G} \tag{12}$$

By performing the partials and time derivatives on the kinetic energy in the Lagrangian the following is obtained,

$$\frac{d}{dt}\left(\frac{\partial KE}{\partial \dot{\vec{Q}}_e}\right) - \frac{\partial KE}{\partial \vec{Q}_e} = M_e\ddot{\vec{Q}}_e + C_e\dot{\vec{Q}}_e \tag{13}$$

where the mass matrix, M_e, is defined by Equation (9) and the velocity matrix, C_e, contains the centrifugal and correolis terms.

By performing the partial of the strain energy terms and the gravitational force, of the Lagrangian, the equations of motion for a single flexible element can be written in the following form:

$$M_e\ddot{\vec{Q}}_e + C_e\dot{\vec{Q}}_e + K_e\vec{Q}_e = \vec{F}_e \tag{14}$$

where, M_e, C_e and K_e is defined by Equations (9), (13) and (11), respectively

Equation (14) is the the differential equations of motion for a single flexible beam element. The equations of motion for the entire system are obtained by assembling the elemental matrices into a global system of equations. This is done through the use of a variable correlation table and constraint functions. These are discussed in the following sections and then the global system of equations are presented.

3.2 VARIABLE CORRELATION TABLE

The elemental matrices developed in the previous section must be assembled to form the governing differential equations of motion which describe the vibration of the entire mechanism. The variable correlation table is used to assemble each element's matrices into the global matrices.

From Equation (14), the vector $\vec{Q}_e$ represents the independent local coordinates of an element which is part of a mechanism. The variable correlation table maps the displacements, $\vec{Q}_e^i$, of the element i into the global displacements $\vec{Q}$, of an entire mechanism. The result is that the general elemental matrices M_e, C_e and K_e, and the forcing vector, $\vec{F}_e$, for each element are assembled into the global vibrational system,

$$M\ddot{\vec{Q}} + C\dot{\vec{Q}} + K\vec{Q} = \vec{F} \tag{15}$$

Assembling M, C, K and $\vec{F}$ in this manner permits the creation of a general algorithm for forming the equations of motion for a general multibody system. This algorithm is applied to each element in turn, until the equations of motion for the entire mechanism have been created.

3.3 CONSTRAINT FUNCTIONS

In this study each element is considered to have both rigid body displacement and flexible deformations. Because of this, maintaining the compatibility between elements is not as straight forward as in a classical Finite Element approach. The variable correlation table can be used to relate one elemental degree of freedom to one global degree of freedom in a classical finite element fashion. This method is sufficient for relating single degrees of freedoms of elements to each other, but, in general it is not sufficient for relating multiply degrees of freedom. In order to handle these situations constraints will be used. Each constraint, Φ, is represented by an equation of the form,

$$\Phi = 0 \tag{16}$$

Position, slope and curvature are the three general types of constraints which are incorporated in this study in order to ensure the compatibility between elements and define the configuration of the mechanism, which can not be handled using the variable correlation table. The specifics of each of the position constraints are discussed in the following subsection.

3.3.1 Position Constraints. Position constraints are used to fix the position of an element's node relative to either time, the ground or the node of another element. Five specific position constraints are employed in this study. The first constraint is a driving constraint which is used to fix a global degree of freedom, $\mathbf{Q}^i$, relative to time. The second time derivative, (i.e. acceleration), is the form employed in this research in order to fix a degree of freedom as a function of time. Specifically,

$$\ddot{\Phi} = \ddot{\mathbf{Q}}^i - qacc(t) = 0 \tag{17}$$

where, $qacc(t)$ is a user defined prescribed function which is assumed to have the form:

$$qacc(t) = a_1 + a_2 t + a_3 t^2 + a_4 \sin(a_5 t) + a_6 \cos(a_7 t) + a_8 \exp(a_9 t) \tag{18}$$

The selection of the the initial position and velocity, $\mathbf{Q}_0^i$ and $\dot{\mathbf{Q}}_0^i$ respectively, and the constants a_i, dictate the exact motion of a single degree of freedom as a function of time. This equation is by no means unique, and was defined so that a wide variety of motions are available. It was defined for this study to offer the properties of polynomial, trigonometric, and exponential terms. The main purpose for this constraint is to define the driving function of a crank link, but it can be applied to any degree of freedom whose position, as a function of time, can be defined by an expression of the form of Equation (18).

If a constant velocity is desired then all of the constants in Equation (18) are set to zero. This equation can also be used to dictate a start-up maneuver in order to achieve a desired operating speed. For example, if a mechanism is starting at rest and is to run at a constant operating speed of θ, and it is desired that this speed be reached in 1 second, then the constants, $a_1, \cdots, a_9$, can be chosen so that $qacc$ function is,

$$qacc(t) = \begin{cases} \theta + \theta\cos(\pi t) & 0.0 \le t < 1.0 \\ 0.0 & t \ge 1.0 \end{cases} \tag{19}$$

This is just one spin-up maneuver which can be used to start a mechanism from rest and bring the operating speed up to a desired constant velocity. By choosing different values for $a_1, \cdots, a_9$ a wide variety of acceleration profiles can be defined to achieve a desired motion.

The second type of position constraint is a ground constraint. This constraint is used to fix the position of an element's node in global space. The ground constraint, which is used in this study, which relates the position of an elements node. $\vec{R}_{PD}$, to the fixed point, $\vec{X}_f$, is given as,

$$\vec{\Phi} = \vec{R}_{PD} - \vec{X}_f = 0 \tag{20}$$

Another type of constraint which relates an element's node to the ground is a sliding constraint. This constraint is used to restrict a node of an element to move along a *sliding* line which is defined by two points in the global workspace, $\vec{X}_1$ and $\vec{X}_2$. This can be done by constraining the coordinate directions of the node, simlar to Equation (20), and/or by using parametric equations to relate the coordinate directions to each other.

If there is a global coordinate direction, i, such that, $\vec{X}_1(i) = \vec{X}_2(i)$, then $\vec{R}_{PD}$ must lie in a plane normal to coordinate direction i, and therefore, $\vec{R}_{PD}(i)$ can be *grounded* in that direction by equating it to the value $\vec{X}_1(i)$. If there are two coordinate directions which meet this criteria, (i.e. $\vec{X}_1(i) = \vec{X}_2(i)$ and $\vec{X}_1(j) = \vec{X}_2(j)$), then $\vec{R}_{PD}$ can be sufficiently constrained to a sliding line. If it is the first node of an element that is sliding, the constraining of a coordinate direction can be accomplished by eliminating the appropriate degree(s) of freedom from the variable correlation table and thus a constraint of this type would not be necessary. If a coordinate direction, i, of $\vec{R}_{PD}$ is to be constrained then, the i^{th} component of Equation (20) can be used, such that,

$$\Phi = \vec{R}_{PD}(i) - \vec{X}_1(i) = 0 \tag{21}$$

For the case of a general sliding constraint, when the variable correlation table and Equation (21) can not sufficiently satisfy the sliding constraint, a set of parametric equations can be defined using the points which define the sliding line, $\vec{X}_1$ and $\vec{X}_2$. These parametric equations relate one coordinate direction to another coordinate direction, and at most 2 equations are required to fully characterize the sliding line. The constants of a resulting parametric equation, $\vec{P}$ and P_0, can be used define a sliding constraint, such that,

$$\Phi = \vec{P}^T\vec{R}_{PD} - P_0 = 0 \tag{22}$$

If Equation (21) can be applied to two coordinate directions, then, $\vec{R}_{PD}$ is sufficiently constrained to be along the *sliding* line, and thus Equation (22) is not needed. Otherwise, if Equation (21) can only be applied once, $\vec{R}_{PD}$ has only been restricted to the plane containing the sliding axis, and must be further restricted by relating the remaining coordinate directions by using Equation (22). For the general case when Equation (21) can not be applied to any of the coordinate directions, Equation (21) must be employed twice in order to sufficiently constrain an elements node to the sliding line.

The previous position constraints relate an element's nodal position to time or the global workspace. A constraint equation which constrains the position of a node of one element to be equal to the position of a node of another element, as illustrated in Figure 2, is given as,

$$\vec{\Phi} = \vec{R}^2_{PD} + T_{GE}T_{e1}\vec{d} - \vec{R}^1_{PD} = 0 \tag{23}$$

where, $\vec{R}^1_{PD}$ and $\vec{R}^2_{PD}$ are the position vectors of the elemental nodes of interest, T_{GE} is the transformation matrix between the global and elemental coordinate systems, T_{e1} is the

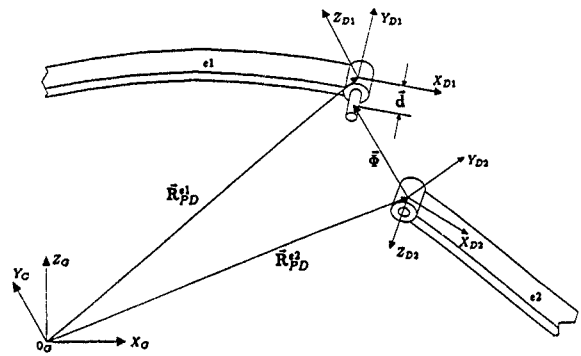


Figure 2: Position Constraint Between Two Deformed Elements.

transformation matrix from the deformed to the undeformed elemental orientation of the node under consideration, and $\vec{d}$ is a fixed offset vector between the two elements relative to the first element's undeformed coordinate system. In general, $\vec{d}$ is used to model the offset which is often present in a pin joint.

The variable correlation table and the position constraint equations presented in this section are sufficient for defining the spatial configuration of most mechanisms. Similar slope and curvature constraints may also be needed in order to fully define the configuration, and insure the compatibility between the various elements of a multibody system.

3.3.2 System of Constraints. The global system of constraints equations, $\vec{\Phi}$, is defined as the total number of constraints required by the configuration and node types of a particular mechanism. Given this system of constraints, in the form of Equation (16), the following must hold for all time,

$$\vec{\Phi} = 0 \tag{24}$$

$$\dot{\vec{\Phi}} = \mathbf{J}\dot{\vec{\mathbf{Q}}} = 0 \tag{25}$$

$$\ddot{\vec{\Phi}} = \mathbf{J}\ddot{\vec{\mathbf{Q}}} + \dot{\mathbf{J}}\dot{\vec{\mathbf{Q}}} = 0 \tag{26}$$

where the constraint jacobian, J, is defined as, $\mathbf{J} = \partial\vec{\Phi}/\partial\vec{\mathbf{Q}}$, and $\vec{\mathbf{Q}}$ are the global degrees of freedom of the mechanism. These constraints are introduced into the global system of equations in the following section using Lagrange multipliers. These equations are subsequently used to eliminate the Lagrange multipliers from the dynamical system of equations for a flexible mechanism.

4 System of Equations of Motion

Up to this point, the differential equations of motion, Equation (14), only applies to a single finite element. The equations of motion for the entire mechanism are determined by assembling the elemental equations into the global equations, using the variable correlation table, and incorporating the constraint jacobian, J, using Lagrange multipliers such that,

$$\mathbf{M}\ddot{\vec{Q}} + \mathbf{C}\dot{\vec{Q}} + \mathbf{K}\vec{Q} + \mathbf{J}^T\vec{\lambda} = \vec{F} \tag{27}$$

where the vector $\vec{Q}$ contains the degrees of freedom of the entire mechanism, and $\vec{\lambda}$ is the vector of Lagrange multipliers corresponding to the constraint functions. By adding the constraint acceleration, Equation (26), this equation can be rewritten as,

$$\begin{bmatrix} \mathbf{M} & \mathbf{J}^T \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\vec{Q}} \\ \vec{\lambda} \end{bmatrix} = \begin{bmatrix} \vec{F} - \mathbf{C}\dot{\vec{Q}} - \mathbf{K}\vec{Q} \\ -\mathbf{J}\dot{\vec{Q}} \end{bmatrix} = \begin{bmatrix} \vec{F}^* \\ -\dot{\mathbf{J}}\dot{\vec{Q}} \end{bmatrix} \tag{28}$$

Solving for $\ddot{\vec{Q}}$ and $\vec{\lambda}$,

$$\begin{bmatrix} \ddot{\vec{Q}} \\ \vec{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M} & \mathbf{J}^t \\ \mathbf{J} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \vec{F}^* \\ -\dot{\mathbf{J}}\dot{\vec{Q}} \end{bmatrix} \tag{29}$$

By computing the inverse of the expanded mass matrix using the inverse by partitions method, or by solving Equation (27) for $\ddot{\vec{Q}}$ and substituting it into Equation (26), it can be shown that,

$$\vec{\lambda} = \left[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right]^{-1}\left[\mathbf{J}\mathbf{M}^{-1}\right]\vec{F}^* + \left[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right]^{-1}\dot{\mathbf{J}}\dot{\vec{Q}} \tag{30}$$

substituting this equation back into Equation (27) and solving for $\ddot{\vec{Q}}$, results in,

$$\ddot{\vec{Q}} = \mathbf{M}^{-1}\vec{F}^* - \left[\mathbf{M}^{-1}\mathbf{J}^T\right]\left[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right]^{-1}\left[\mathbf{J}\mathbf{M}^{-1}\right]\vec{F}^* - \left[\mathbf{M}^{-1}\mathbf{J}^T\right]\left[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T\right]^{-1}\dot{\mathbf{J}}\dot{\vec{Q}} \tag{31}$$

This system of second order ordinary differential equations of motion satisfies the constraints without the need to include the Lagrange multipliers, thereby not increasing the size of an already large problem. This technique will be used to solve for the elastic deformations as well as the rigid body motions of a general flexible mechanism or multibody system.

4.1 SOLUTION TECHNIQUE

In the previous sections, the general formulation of the global system of equations of motion for a general flexible mechanism was presented. The matrices M, C, K and the forcing vector $\vec{F}$ in Equation (31) are nonlinear. Because these equations are nonlinear and the transient solution is desired, numerical techniques are used to solve the global system of equations.

Equation (31) is the system of second order ordinary differential equations which will be solved. But, in order to solve these equations, they must first be rewritten as a set of

first order ordinary differential equations which can then be numerically integrated by most standard ODE solvers. This is done by defining $\bar{V}$ such that,

$$\dot{\bar{V}} = \begin{bmatrix} \dot{\bar{v}} \\ - - - \\ \bar{v} \end{bmatrix} = \begin{bmatrix} \left(M^{-1}\bar{F}^{*} - \left[M^{-1}J^{T}\right]\left[JM^{-1}J^{T}\right]^{-1}\left[JM^{-1}\right]\bar{F}^{*}\right. \\ \left. - \left[M^{-1}J^{T}\right]\left[JM^{-1}J^{T}\right]^{-1}\dot{J}\dot{\bar{Q}}\right) \\ - - - - - - - - - - - - - - - - - - \\ \dot{\bar{Q}} \end{bmatrix} \quad (32)$$

This equation was solved using the LSODE routine from the ODEPACK library. LSODE uses a 12^{th} order Adams-Moulton method to solves a system of ordinary differential equations.

5 Rigid Double Pendulum Example

A driven crank rigid double pendulum example is presented in this section to illustrate the position constraint functions, and solution technique presented in this paper. Figure 3 illustrates the rigid double pendulum model for this example. Each link of the pendulum

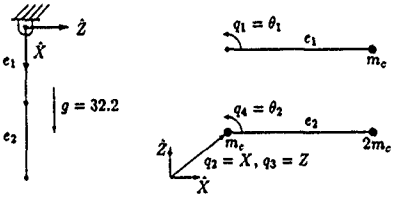


Figure 3: Rigid Double Pendulum Model.

Length of the links	l_1 and l_2	1.00
Cross Sectional Area	A	0.10
Mass Density	ρ	1.00
Concentrated Mass, m_c	m	0.10

Table 1: Geometric Properties of the Rigid Double Pendulum.

is considered to be a separate element. Both of these elements have a rigid body rotational degree of freedom, θ_1 and θ_2, and the second link has rigid body position degrees of freedom, X and Z. The rigid body position degrees of freedom of the second link are equated to the position of the second node of the first element using a position constraint. This model also has concentrated masses at the nodes and is subjected to a gravitational force, as indicated in Figure 3. Other than the concentrated masses, both links have identical material and geometric properties, as shown in Table 1.

The exact second order nonlinear ordinary differential equations of motion for a free vibrating double pendulum are given in [25,26]. For this example $m = m_1 = m_2 = 0.1$, $l_1 = l_2 = 1.0$, and $m_{c1} = m_{c2} = 2m$. By substituting these values for the link lengths and masses, into the equations of motion for a double pendulum, the exact system of equations of motion for the rigid double pendulum studied in this example is defined as,

$$\begin{bmatrix} \frac{16}{3}m\ddot{\theta}_1 + \frac{4}{2}m\cos(\theta_2-\theta_1)\ddot{\theta}_2 - \frac{4}{2}m\sin(\theta_2-\theta_1)\dot{\theta}_2^2 + \frac{11}{2}mg\sin\theta_1 \\ \frac{4}{3}m\ddot{\theta}_2 + \frac{4}{2}m\cos(\theta_2-\theta_1)\ddot{\theta}_1 + \frac{4}{2}m\sin(\theta_2-\theta_1)\theta_1^2 + \frac{4}{2}mg\sin\theta_2 \end{bmatrix} = 0 \quad (33)$$

The solution of these equations is compared to the solution obtained from the program developed in this study. For this example the pendulum was given a driving function. The driving function was a constant angular velocity, applied to the first link of the pendulum, θ_1. The initial position of the pendulum was $\theta_1 = \theta_2 = 0$ and the links were given initial velocities of $\theta_1 = 10 rad/sec$ and $\theta_2 = 0 rad/sec$, respectively. The constant angular velocity for θ_1 was achieved by applying an acceleration constraint of $\ddot{\theta}_1 = 0$ over the entire time duration of the simulation. This example was run for a 3.0 second duration An integration tolerance of 1.0E-08 was used and 3001 data points were obtained at intervals of 0.001 seconds in 5.08 CPU seconds, on an IBM RISC System/6000 Model 550 workstation running AIX Version 3.1.

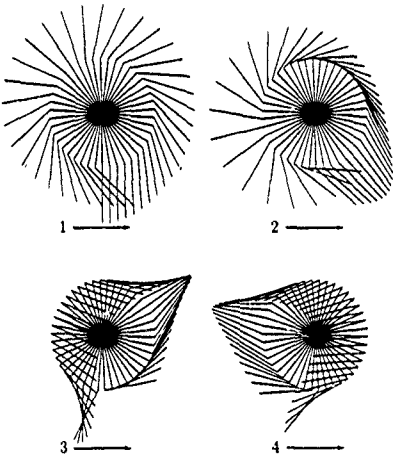


Figure 4: Driven Rigid Double Pendulum Time History.

Figure 4 shows the time history for the first four complete revolutions. The positions illustrated in this figure are at intervals of 0 015 seconds In this figure, the numbers 1, 2, 3, and 4 indicates the consecutive revolutions and the arrows indicate the direction of

motion. The first two revolutions shown in this figure clearly illustrate the effect gravity has on the system. The second link lags behind the first link on the up-swings and leads on the down-swings. The third and fourth revolutions show the full effects of this as the second link folds back on the first link. This is further illustrated in Figure 5, which is a plot of the endpoint of the pendulum. This plot clearly shows the *stall* points in the third and forth revolutions, and the second link folding back on the first link Given the initial conditions for this example, the results from Equation (33) are identical to those illustrated in Figures 4, and 5.

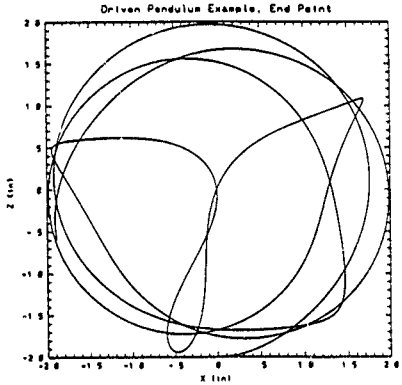


Figure 5: Driven Rigid Double Pendulum Endpoint Results.

The rigid double pendulum example presented in this section shows that the rigid body aspects of the formulation presented in this study are exact. It also shows that the use of the position and acceleration constraints is viable for rigid models. More importantly this example shows that the Lagrange Multipliers can be solved for directly and substituted back into the original dynamical system of equations, without any loss of data or accuracy, and subsequently decreasing the size of the problem to be solved. This same pendulum is now considered to be flexible in order to show that these statements are valid for systems containing rigid as well as flexible modes of vibration.

6 Flexible Double Pendulum

In this section the model used for the rigid double pendulum example was considered to be flexible. The example presented in this section will verify the developed methodology's capability to handle models which contain both flexible and rigid modes. The model used for the rigid double pendulum example, Figure 3, is given flexible degrees of freedom and material properties in this case, which would promote large deformations.

Figure 6 illustrates the flexible double pendulum model which is an extension of the rigid model, and, as with the rigid example, this example is also subjected to a gravitational force. The physical and material properties for this flexible pendulum example are listed in Table 2

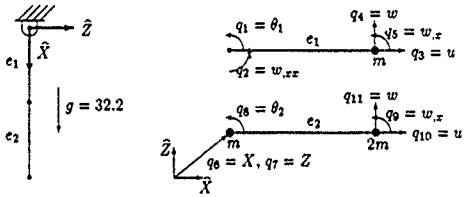


Figure 6. Flexible Double Pendulum Example Model.

Length of the links	l_1 and l_2	1.00
Cross Sectional Area	A	0.10
Moments of Inertial	I_{yy} I_{zz}	1.0E-02
Mass Density	ρ	1.00
Concentrated Mass	m	0.10
Modulus of elasticity	$E(lb/in^2)$	1.0E+04
Shear modulus	$G(lb/in^2)$	1.0E+05

Table 2: Geometric and Material Properties for the Flexible Pendulum.

This example was run with a spin-up maneuver which result in $\theta_1 = 10.0 rad/sec$ for $t \geq 1.0 sec$. The pendulum started at rest, with the axial deformations equal to the static deformations which result under the present gravitational force and an initial position of $\theta_1 = \theta_2 = 0$ radians.

The spin-up maneuver for this example was accomplished by specifying an acceleration constraint of,

$$\ddot{\theta}_1 = \begin{cases} 10.0 + 10.0\cos(\pi t) & 0.0 \leq t < 1.0 \\ 0.0 & t \geq 1.0 \end{cases} \tag{34}$$

Given this constraint equation and initial conditions, this case was run for 6.0 seconds. An integration tolerance of 1.0E-06 was used and 6001 data points were obtained at intervals of 0.001 seconds in 2948.18 CPU seconds.

Figures 7 and 8 shows the time history, at intervals of 0.015 seconds, for the first eight revolutions of the pendulum. The consecutive revolutions and the direction of rotation are indicated by 1, ···, 8 and the arrows. Figures 9 and 10 show the axial and transverse deformations, respectively, of the elements In both of these plots the solid lines are the results for element 1 and the dashed lines are the results for element 2. These figures

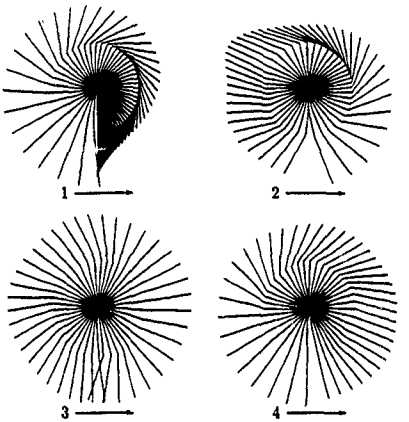


Figure 7: Driven Flexible Pendulum Example, Time History 1-4 Revolutions.

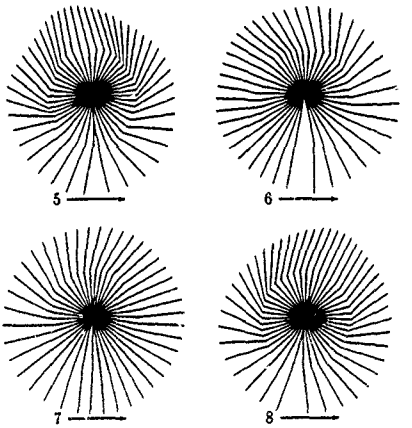


Figure 8: Driven Flexible Pendulum Example, Time History 5-8 Revolutions.

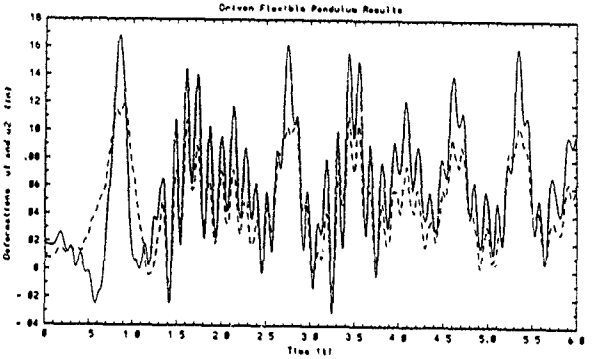


Figure 9: Driven Flexible Pendulum Example, Axial Deformations, u.

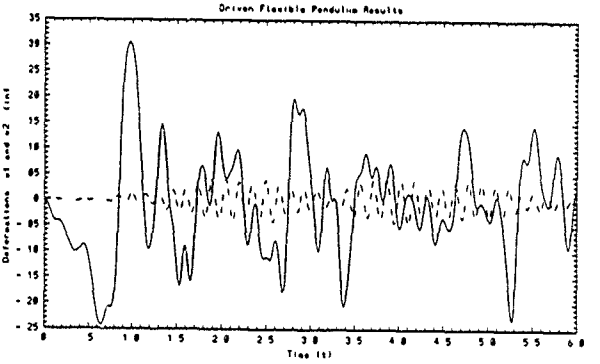


Figure 10: Driven Flexible Pendulum Example, Transverse Deformations, w.

shows that the first link is subjected to axial and transverse deformations of 15% and 30% of the link length, respectively. As is expected, the deformations of the first element are, in general, larger than those of the second element. The axial deformations are larger because the first link is subjected to the additional centrifugal force because of the mass of the second link. The transverse deformations are higher than those of the the second link because the first element is subjected to the torque required to cause the prescribed motion, and thus an additional moment at the base.

This example illustrates that the methodology developed in this study is more than capable of handling problems which contain both flexible and rigid body motion. It also shows that the position constraint functions work when flexible and rigid body modes are present. The examples presented up to this point show the that model developed in this study is, in theory, valid. Now an example will be presented which is compared to experimental results in order to verify the developed model's capability to accurately model the response of actual high-speed flexible mechanisms.

7 Planar Four-Bar Mechanism

The example presented in this section is a four-bar mechanism whose experimental results are given by Sutherland in [27]. This four-bar mechanism is illustrated in Figure 11. The results obtained from the developed program are compared to experimental results in this section. This was done in order to verify the capability of the methodology presented in this study to accurately model the response of high-speed flexible mechanisms.

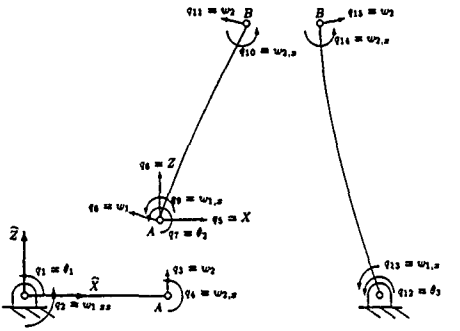


Figure 11: Planar Four-Bar Mechanism Model.

Figure 11 illustrates the fully flexible four-bar mechanism which contains all of the relevant rigid body and flexible degrees of freedom. The physical and material properties of this mechanism are listed in Table 3. For the case presented in this section the mechanism was started from rest with all deformations equal to zero and an initial crank angle of $\theta_1 = 0$ radians. The actual operating speed used in the experiment was not known for this problem.

	Crank	Coupler	Follower
Modulus of Elasticity $[E(lb/in^2)]$	15 00E+06	15 00E+06	15.00E+06
Mass Density $[\rho(lb\,s^2/in^4)]$	7 920E-04	7.920E-04	7.920E-04
Link Area $[A(in^2)]$	2.344E-02	2.344E-2	2.344E-02
Moment of Inertial $[I(in^4)]$	3.827E-05	3 827E-05	3.827E-5
Link Length $[L(in)]$	4 064E+00	1.000E+01	1.000E+01
Mass at Node 1 $[m_{con1}(lbm)]$	0.000E+00	1.0 .0E-05	0 000E+00
Mass at Node 2 $[m_{con2}(lbm)]$	1 000E-05	1 000E-05	1.000E-05

Length of the base link = 13.55 in.

Table 3: Geometric Properties for the Flexible Four-Bar Mechanism.

By using the information available in [27] the operating speed, $\dot{\theta}$, was determined to be in the range $23 \leq \theta \leq 26$ *rad/sec*. This example was run at various operating speeds in this range. In order to achieve a desired operating speed a spin-up maneuver was performed. Then the mechanism was run at the desired constant velocity The spin-up maneuver for this example was accomplished by specifying an acceleration constraint of,

$$\ddot{\theta}_1 = \begin{cases} \theta + \theta\cos(\pi t) & 0.0 \leq t < 1.0 \\ 0.0 & t \geq 1.0 \end{cases} \tag{35}$$

In order to promote stability and simulate material damping, a nominal damping factor of $\xi = 0.000002$ was used for each case. Given the damping coefficient and the initial conditions this four-bar mechanism was run at operating speeds of $\theta = 22, 23, 24, 25, 26,$ and 27 *rad/sec*. Each case was run for a time which corresponded to 10 full revolutions, after the completion on the spin-up maneuver. An integration tolerance of 1 0E-06 was used and 500 data points were obtained for each full revolution at intervals corresponding to the operating speed. The average CPU time for these cases was 1100 CPU seconds

The problem reached a *steady-state* response after 3-4 revolutions. The operating speed was properly bracketed to be in the 24-25 *rad/sec* range. Figure 12 shows the results for $\theta = 24$ In this figure the solid line is the for the fifth revolution and the dashed line is the experimental results. This figure shows that the results compare extremely favorably with the experimental results, especially considering the fact that all the links are flexible and are undergoing large deformations.

This example verifies that the methodology which has been developed during this study is capable of handling models which contain flexible and rigid body motion It also shows that the developed methodology can accurately model the nonlinear response of high-speed flexible mechanisms with just a few elements. The solution technique for handling constraints, presented in this paper, has also proven to be a good alternative to the classical

methods for modeling constrained multibody systems.

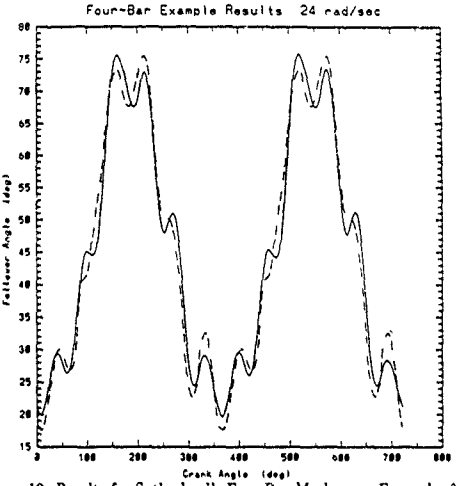


Figure 12: Results for Sutherland's Four-Bar Mechanism Example, $\theta = 24$

8 Conclusions

The examples presented in this paper, and those given in reference [25], show that the technique developed in this research is capable of determining the exact rigid body motion of a mechanical system. It is also able to calculate large deformations combined with rigid body motion. This was verified through comparisons with both analytical and experimental results of others. More importantly the technique for handling constraints presented in this paper has proven to be an effective alternative to the classical methods for modeling constrained multibody systems. Although the method involves determining the inverse of the mass matrix M, and the matrix $[\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T]$, these inverses can be calculated very efficiently since the mass matrix is symmetric and normally quite sparse. This method also avoids having to calculate the Lagrange multipliers and the constraint equations.

References

[1] Bahgat, B. M. and K. D. Willmert, *'Finite Element Vibrational Analysis of Planar Mechanisms,'* Mechanisms and Machine Theory, Vol. 11, 1976, pp 47-71.

[2] Hughes, P. C., *'Dynamics of a Chain of Flexible Bodies,'* Journal of the Astronautical Sciences, Vol. 27, No. 4, Oct-Dec. 1979, pp. 359-380.

[3] Masurekar, V. and K. N. Gupta, *'Theoretical and Experimental Kineto Elastodynamic Analysis of High Speed Linkage,'* Mechanism and Machine Theory, Vol. 24, No. 5, 1989, pp. 325-334.

[4] Yoo, W S. and E. J. Haug, *'Dynamics of Articulated Structures. Part 1. Theory,'* Journal for Structural Mechanics, Vol. 14 No. 1, 1986, pp. 105-126.

[5] Yoo, W. S. and E. J. Haug, *'Dynamics of Articulated Structures. Part 2. Computer Implementation and Applications,'* Journal for Structural Mechanics, Vol. 14 No. 2, 1986, pp. 177-189.

[6] Rosen, A., R. G. Loewy and M. B. Mathew, *'Nonlinear Dynamics of Slender Rods,'* AIAA Journal, Vol. 25, No. 4, April 1987, pp. 611 619.

[7] Bartolone, D. F. and A. A. Shabana, *'Effect of Beam Initial Curvature on the Dynamics of Deformable Multibody Systems,'* Mechanisms and Machine Theory, Vol. 24, No. 5, 1989, pp. 411-429.

[8] Liou, F. W., and A. G. Erdman, *'Analysis of a High-Speed Flexible Four-Bar Linkage: Part 1 — Formulation and Solution,'* Journal of Vibration, Acoustics, Stress, and Reliability in Design, Trans. ASME, Vol. 111, Jan. 1989, pp 35-41.

[9] Liou, F. W., and A. G. Erdman, *'Analysis of a High-Speed Flexible Four-Bar Linkage: Part 2 — Analytical and Experimental Results on the Apollo,'* Journal of Vibration, Acoustics, Stress, and Reliability in Design, Trans. ASME, Vol. 111, Jan. 1989, pp. 42-47.

[10] Tennant, D. W., K. D. Willmert and M. Sathyamoorthy, *'Finite Element Nonlinear Vibrational Analysis of Planar Mechanisms.'* Material Nonlinearity in Vibrational Problems, AMD-Vol. 71, pp. 79-89.

[11] Bauchau, O. A and C. H. Hong, *'Nonlinear Response and Stability Analysis of Beams Using Finite Elements in Time,'* AIAA Journal, Vol. 26, No 9, Sept. 1988, pp. 1135-1142.

[12] El-Sawy, E. E., K. D. Willmert, and M. Sathyamoorthy, *'Nonlinear Vibrations of Three-Dimensional Linkages,'* To appear in Mechanisms and Machine Theory.

[13] Jerkovsky, W., *'The Structure of Multibody Dynamics Equations,'* Journal of Guidance and Control, Vol. 1, No. 3, May-June 1978, pp. 173-182.

[14] Kamman, J. W. and R. L. Huston, *'Constrained Multibody System Dynamics,'* Computers & Structures, Vol. 18, No. 6, 1984, pp. 999-1003.

[15] Kane, T. R. and D A. Levinson, *'Dynamics: Theory and Applications,'* McGraw-Hill Book Company, 1985.

[16] Manko, D. J. and W L. Whittaker, *'Inverse Dynamic Models of Closed-Chain Mechanisms with Contact Compliance,'* Journal of Mechanical Design, Trans ASME, Vol. 114, Mar. 1992, pp. 82-86.

[17] Singh, R. P., R. J. VanderVoort and P. W. Likins, *'Dynamics of Flexible Bodies in Tree Topology — A Computer-Oriented Approach,'* Journal of Guidance and Control, Vol. 8, No 5., Sept.-Oct. 1985, pp. 584-590.

[18] Ider, S. K. and F. M. L. Amirouche, *'Nonlinear Modeling of Flexible Multibody Systems Dynamics Subjected to Variable Constraints,'* Journal of Applied Mechanics, Trans ASME, Vol. 56, June 1989, pp. 444-450.

[19] Bayo, E., J. Garcia De Jalon and M. A. Serna, *'A Modified Lagrangian Formulation for the Dynamic Analysis of Constrained Mechanical Systems,'* Computer Methods in Applied Mechanics and Engineering, Vol. 71, Nov. 1988, pp 183-195.

[20] Amirouche, F. M. L. and F. L. Huston, *'Dynamics of Large Constrained Flexible Structures,'* Journal of Dynamic Systems, Measurement and Control, Trans. ASME, Vol. 110, Mar. 1988, pp. 78-83.

[21] Kim, S. and E. J. Haug, *'A Recursive Formulation for Flexible Multibody Dynamics, Part I: Open-Loop Systems,'* Computer Methods in Applied Mechanics and Engineering, Vol. 71, Dec. 1988, pp. 293-314.

[22] De Smet, M., C. Liefooghe, P. Sas and R. Snoeys, *'Dynamic Analysis of Flexible Structures Using Component Mode Synthesis,'* Journal of Applied Mechanics, Trans. ASME, Vol. 56, Dec. 1989, pp. 874-880.

[23] B. Samanta, *'Dynamics of Flexible Multibody Systems Using Bond Graphs and Lagrange Multipliers,'* Journal of Mechanical Design, Trans. ASME, Vol. 112, Mar. 1990, pp. 30-35.

[24] Lankarani, H. M. and P. E. Nikravesh, *'Canonical Impulse-Momentum Equations for Impact Analysis of Multibody Systems,'* Journal of Mechanical Design, Trans. ASME, Vol. 114, Mar. 1992, pp. 180-186.

[25] Russell, D. M. *'The Transient Response of Geometrically and Materially Nonlinear Spatial Mechansims,'* Ph.D. Dissertation, Clarkson University.

[26] Meirovitch, L., *'Elements of Vibration Analysis,'* 2nd. Edition, McGraw-Hill, New York, 1986.

[27] Sutherland, G. H., *"Analytical and Experimental Investigation of a High-Speed Elastic-Membered Linkage,'* Journal of Engineering for Industry, Trans. ASME,Vol 98, No. 3, Aug. 1976, pp. 788-794

EIGENVALUE ANALYSIS IN FLEXIBLE MULTIBODY DYNAMICS BY USE OF THE NON-SYMMETRIC LANCZOS ALGORITHM

D B. DOAN
M.GERADIN
LTAS - University of Liège
21, rue Ernest Solvay
4000 Liège, Belgium

N. KILL
SAMTECH S.A.
Bld. Frère Orban, 25
4000 Liège, Belgium

ABSTRACT. This paper presents the use of the non-symmetric Lanczos algorithm in order to solve the eigenvalue problem of a flexible mechanism around a determined configuration. The algorithm is implemented into the MECANO software which aims at investigating the behavior of flexible and rigid mechanisms using the finite element approach.

1. Introduction

In the context of multibody dynamics, the eigenvalue analysis is mainly used to investigate the stability of the physical system. The use the finite element method involves the handling of large sparse system matrices. When the geometric non-linearity and physical phenomena such as gyroscopic and dry friction effects are taken into account, the assumption of symmetric positive definite system matrices is no longer valid. The use of a general algorithm for solving the eigenvalue problem is then required. The objective of the paper is to propose such an algorithm. The first part of the paper briefly presents some general aspects of the MECANO software in which the algorithm is implemented. The second part mainly deals with the algorithmic aspects of the non-symmetric Lanczos algorithm. The last part presents some preliminary results of the implementation.

2. General Aspects of the MECANO Software ([5], [6])

The MECANO software is a specific module of the general purpose finite element code SAMCEF. Based on the finite element concept, it provides an answer to industrial requirements in the field of flexible articulated system analysis.

The mechanism is described in terms of absolute coordinates. The flexible members, rigid links and mechanical joints are considered as individual elements of a large library of components. The topology of the mechanism is thus implicitly contained in the finite element mesh describing the system. This method of modeling is not penalized by multi-connected topology or closed kinematic loops and provides a simple and clear data set Integrated in a larger finite element concept by the substructuring technique, it allows to model any type of flexible link with arbitrary shape. Making use of the *customized function* and/or the *user element*, the software also allows to model excitations and/or components which are not standard in the element library.

The system of motion equations is derived from Hamilton's principle and assembled in sparse form in order to handle efficiently very large models. The augmented Lagrangian method takes into account the constraints on the system. The augmented Lagrange's functional is described as follows :

$$\mathcal{L}^* = \mathcal{L}(q) - k\lambda^T\phi + \frac{p}{2}\|\phi\|^2 \qquad (2.1)$$

where

- $\mathcal{L}$ is the lagrangian of the mechanism,
- q is the generalized coordinate vector described in the absolute frame,
- ϕ is the constraint vector and λ the Lagrange's multipliers,
- k and p are respectively scaling and penalty factors. They vary in terms of the generalized stiffness of the mechanism. The application of Hamilton's principle to (2.1) yields to the following DAE system :

$$\begin{cases} M\ddot{q} + B^T(k\lambda + p\phi) = g(q,\dot{q},t) \\ k\phi(q,t) = 0 \end{cases} \qquad (2.2)$$

If $(q^*, \dot{q}^*, \ddot{q}^*, \lambda^*)$ is an approximated solution of the system (2.2) at time t, expressing the correction at $(t + \Delta t)$

$$(q^* + \Delta q, \dot{q}^* + \Delta\dot{q}, \ddot{q}^* + \Delta\ddot{q}, \lambda^* + \Delta\lambda)$$

allows to obtain the linearized system of equations in the form :

$$\begin{cases} M\Delta\ddot{q} + C\Delta\dot{q} + S\Delta q + kB^T\Delta\lambda = r^* \\ kB\,\Delta q = -k\phi^* \end{cases} \qquad (2.3)$$

where M, C and S are respectively the mass, damping and stiffness tangent matrices. They are sparse matrices assembled under skyline form and are also unsymmetric due to the gyroscopic effects and dry friction forces. M, C and S depend on the generalized coordinates and on their derivatives.

3. Non-symmetric Lanczos Algorithm

3.1. GENERAL PROBLEM

The general solution of the system of equations

$$M\ddot{q} + B\dot{q} + Kq = 0 \tag{3.1}$$

can be expressed as

$$q(t) = qe^{pt} \quad \text{with} \quad p = \alpha + i\omega$$

where α is defined as the damping coefficient and ω is the natural frequency. In terms of the state vector $[q \;\; pq]$, equation (3.1) may be rewritten as a first order system of equations

$$\begin{bmatrix} -B & -M \\ -M & O \end{bmatrix} \begin{bmatrix} q \\ pq \end{bmatrix} = \frac{1}{p} \begin{bmatrix} K & O \\ O & -M \end{bmatrix} \begin{bmatrix} q \\ pq \end{bmatrix} \tag{3.2}$$

The standard form of the eigenvalue is then obtained

$$\mathcal{A}x = \lambda \mathcal{B}x \quad \text{with} \quad \lambda = \frac{1}{p} \tag{3.3}$$

When all the kinematic and parasitic rigid modes are filtered out, matrix B^{-1} exists and equation (3.3) is then expressed as follows

$$\mathcal{C}x = \lambda x \quad \text{with} \quad \mathcal{C} = B^{-1}\mathcal{A} \tag{3.4}$$

Since $\mathcal{C}$ is unsymmetric, the eigenvalue problem admits different eigensolutions to the left denoted z and to the right denoted x. Orthogonality of eigensolutions is expressed between z and x. The general problem and its orthogonality conditions can be written as follows

$$\begin{cases} \mathcal{C}x = \lambda x & \text{the right-hand problem} \\ \mathcal{C}^T z = \lambda z & \text{the left-hand problem} \end{cases} \tag{3.5}$$

with the orthogonality relationships

$$\begin{cases} Z^T \mathcal{C} X = J \\ Z^T X = \Lambda J \end{cases} \tag{3.6}$$

where Λ is a diagonal matrix collecting the eigenvalues λ_i and J is a signed unit matrix (± 1 on the diagonal). X and Z contain the eigenvectors of the problem

3.2 NON-SYMMETRIC LANCZOS ALGORITHM ([1],[7])

The non-symmetric Lanczos algorithm is a generalization of the well-known Lanczos symmetric scheme. The main advantage arises from the fact that the orthogonal sequences of iterates are built up from iterations on only one left-hand and one right-hand iterate at a time. In the general case, the algorithm may be formulated as follows

a. *Choice of right- and left-hand starting vectors* $\mathbf{d}$ *and* $\mathbf{g}_0$ in the form

$$\mathbf{d}_0 = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \quad \text{and} \quad \mathbf{g}_0 = \mathcal{B} \begin{bmatrix} v_0 \\ u_0 \end{bmatrix} \tag{3.7}$$

where u_0 and v_0 are arbitrarily generated. In order to satisfy the constraints of the system, it is necessary to perform one power iteration on $\mathbf{g}_0$ and $\mathbf{d}_0$ before the beginning of the Lanczos iteration.

b. *Power iteration :* the inverse iterations are then performed on both left- and right-hand vectors

$$\overline{\mathbf{d}}_{k+1} = \mathcal{C}\mathbf{d}_k \quad \text{and} \quad \overline{\mathbf{g}}_{k+1} = \mathcal{C}^T \mathbf{g}_k \tag{3.8}$$

c. *Biorthonormalization of* $\mathbf{g}$ *and* $\mathbf{d}$

$$\begin{cases} \gamma_{k+1}\mathbf{d}_{k+1} = \overline{\mathbf{d}}_{k+1} - \alpha_k \mathbf{d}_k - \beta_{k-1}\mathbf{d}_{k-1} = \mathbf{d}^*_{k+1} \\ \gamma_{k+1}\mathbf{g}_{k+1} = \overline{\mathbf{g}}_{k+1} - \alpha^*_k \mathbf{g}_k - \beta^*_{k-1}\mathbf{d}_{k-1} = \mathbf{g}^*_{k+1} \end{cases} \tag{3.9}$$

The orthonormality condition between the left- and the right-hand iterates allows to compute the coefficients

$$\begin{cases} \mathbf{d}_{k+1}{}^T \mathbf{g}_j = 0 \\ \mathbf{g}_{k+1}{}^T \mathbf{d}_j = 0 \\ \text{for} \quad j < k \end{cases} \Rightarrow \begin{cases} \alpha_k = \alpha^*_k = e_k \mathbf{g}_k{}^T \mathcal{C} \mathbf{d}_k \\ \beta_k = \beta_k{}^* = \gamma_k e_k e_{k-1} \\ \gamma_k = \sqrt{|\mathbf{g}_k{}^{*T} \mathbf{d}_k{}^*|} \\ e_k = \mathbf{g}_k{}^T \mathbf{d}_k = \pm 1 \end{cases} \tag{3.10}$$

Under matrix form, equation (3.9) is then expressed as

$$\begin{cases} C D_k = D_k T_k + (0,\ldots,0,d_{k+1}^*) \\ C^T G_k = G_k T_k + (0,\ldots,0,g_{k+1}^*) \end{cases} \tag{3.11}$$

where T is the non-symmetric tridiagonal matrix

$$T = \begin{bmatrix} \alpha_0 & \beta_0 & 0 & 0 & \cdots & \cdots & 0 \\ \gamma_1 & \alpha_1 & \beta_1 & 0 & \ddots & \cdots & 0 \\ 0 & \gamma_2 & \alpha_2 & \beta_2 & 0 & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & 0 & \ddots \\ \vdots & 0 & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & & \ddots & \ddots & \beta_{k-1} \\ 0 & \cdots & \cdots & \cdots & 0 & \gamma_k & \alpha_k \end{bmatrix}$$

d. *Solving the interaction problem*

The interaction eigenvalue problem is then written in the form

$$J_k T_k r = \lambda J_k r \quad \text{and} \quad T_k^T J_k s = \Lambda J_k s \tag{3.12}$$

with the matrix

$$J_k = diag(e_k)$$

The HQR algorithm allows then to perform the eigenvalue analysis of (3.12). The left- and right-hand eigenvectors of the original problem are then restored in the form

$$x = D_k r \quad \text{and} \quad z = G_k s \tag{3.13}$$

The associated Rayleigh quotient is computed in the form

$$\lambda = \frac{z^T C x}{z^T x}$$

and provides very fast convergence to the eigenvalues of C.

The iteration procedure is stopped when either the allowed number of iterations is reached or the convergence criterion is satisfied ([7]). During the iterative procedure, the orthogonality between the left and right sequences is rapidly lost. It is thus necessary to use the orthogonalization procedure of Gram-Schmidt in order to recover the orthogonality which increases significantly the computational cost of the algorithm.

Let us finaly note that in case of multiple eigenvalues, the Lanczos algorithm allows to obtain only one eigenmode at a time. the extraction of multiple eigenvalues can not be performed without a restarting of the algorithm. The new starting vectors d_0^1 and g_0^1 are chosen orthogonal to the previously converged iterates. The restart procedure ought to be performed several times in order to extract all the eigenmodes of a multiple frequency [7].

3 3. FILTERING OF RIGID BODY MODES AND PARASITE MODES

The existence of rigid body and kinematic modes is indicated by a rank reduction of the matrix B. Two solutions are proposed for solving this problem to perform the eigenvalue analysis in the presence of singularities.

3.3.1 Filtering Operator. This method is elegant and powerful but requ're the a priori determination of the rigid modes U. The filtering operator $\mathcal{P}$ is then built as follows .

$$\mathcal{P} = \mathrm{I} - \mathrm{U}\mathrm{U}^T\mathcal{M} \quad \text{with} \quad \mathcal{M} = \begin{bmatrix} \mathrm{M} & 0 \\ 0 & 0 \end{bmatrix}$$

The rigid body modes are filtered out if the orthogonality condition is respected

$$\begin{cases} \mathrm{U}^T\mathrm{M}\mathrm{D}_1 = \mathrm{O} & \text{for the right-hand problem} \\ \mathrm{G}_1^T\mathrm{U} = \mathrm{O} & \text{for the left-hand problem} \end{cases} \tag{3.14}$$

where the subscript $(_1)$ denotes the real part of the iterates. The condition (3.14) is fulfilled by use of Gram-Schmidt orthogonalization procedure. The computation of U is however difficult when the skyline form of the tangent matrices must be respected.

3.3.2 Positive Shifting. If p_s is a shift parameter, the solution $p_1 = p + p_s$ allows to express the characteristic equation (3.1) as

$$\mathrm{M}(q)\ddot{q} + \mathrm{B}^*(q)\dot{q} + \mathrm{K}^*(q)q = 0 \tag{3.15}$$

where

$$\mathrm{B}^* = \mathrm{B} + 2p_s\mathrm{M},$$

$$\mathrm{K}^* = \mathrm{K} + p_s\mathrm{B} + p_s^2\mathrm{M}$$

This method is simple and efficient, it does not alter the skyline form of the M, C and K matrices Nevertheless, the choice of the shift parameter p_s remains difficult and "jump" may occur over some eigenfrequencies.

At the present time, only the positive shifting procedure has been implemented in the algorithm

4. Stability Analysis of a Helicoidal Spring

The system to be analyzed is a helicoidal spring designed to start a turbopump (figure 4.1). The cross section of the wire is rectangular. The upper and the lower sides of the spring are constrained on two parallel planes whose distance is fixed. When the spring is loaded by a torque C onto its upper end, the main deformation of the turn implies coupling between bending and torsion and induces strong geometric nonlinearities. The goal of the analysis to determine the buckling load C_B for which the spring turns are warped.

The spring is modeled by 37 beam elements with rectangular cross section It is made of steel. The total number of degrees of freedom is 222. Figure 4.2 presents the finite element model of the system. The lower end of the spring is clamped and a torque is applied onto the upper end in the axial direction. The incremental static analysis is performed until buckling occurs. When the spring is warped, its turns are twisted and the analysis diverges by numerical instability. In order to determine precisely the buckling torque, the load increment is refined at the end of the analysis.

An eigenvalue analysis is performed after every 5 increments to determine the residual stability of the system. Buckling can be detected as the moment when the first vibration frequency of the system becomes zero.

Figure 4.3 shows an intermediate configuration and the buckled one. Figure 4.4 displays the rotation angle versus the applied torque, showing that the spring response is linear up to bifurcation. Figure 4.5 shows the evolution of the first five eigenfrequencies versus applied torque C. Under external load, the system is very sensitive to nonlinear geometric effects and one notes the contineous decrease of the first eigenfrequency with the torque. A rapid decrease is observed just befor the occurence of the buckling phenomenon

5. Conclusion

The non-symmetric Lanczos algorithm is an efficient tool to investigate the stability of flexible mechanisms. It permits to take into account the influence of the mechanical effects introduced by the modeling of the gyroscopic effect and of the dry friction. Improvements such as an acceleration technique by use of Chebychev polynomials [3] would still render it more attractive. The algorithm is still under test on more general examples.

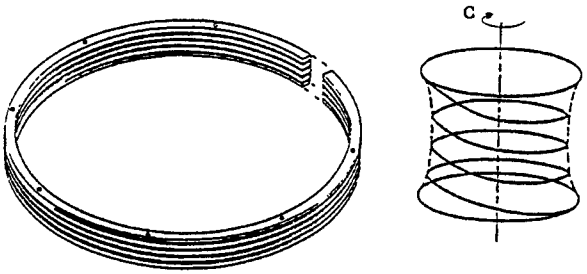


Figure 4.1 . the helicoidal spring

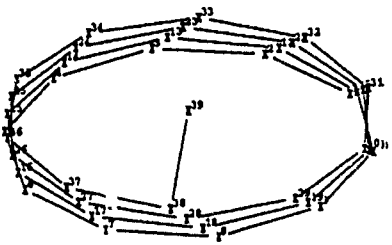
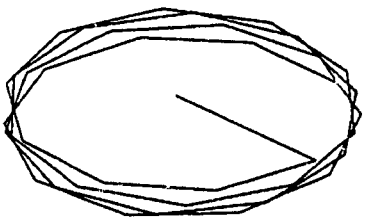
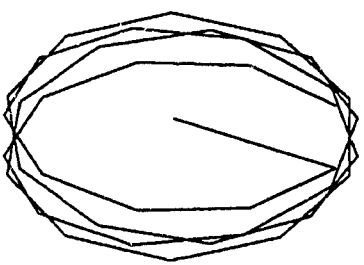


Figure 4 2 : Finite element model of the spring



Regularly deformed configuration before buckling



Buckled configuration

Figure 4.3 : Two configurations of the spring

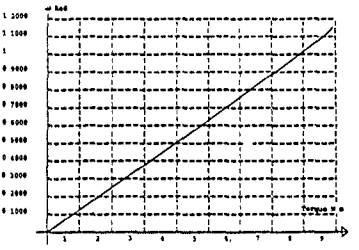


Figure 4.4 . Evolution of the rotation angle in terms of the applied torque

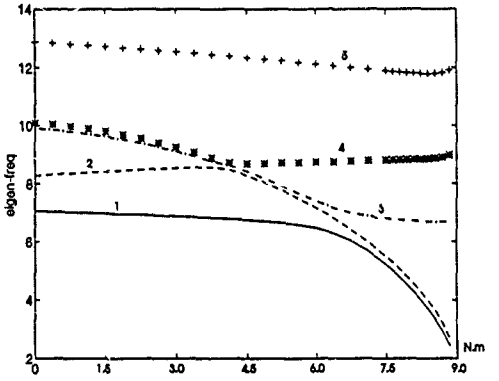


Figure 4.5 · Evolution of first five eigenvalues in terms of the applied torque

6. References

1 F CHATELIN, *Valeurs Propres de Matrices*, chap. 6, pp. 180-188, ed. Masson, 1988.

2 Y SAAD, *Partial Eigensolutions of Large Nonsymmetric Matrices*, Report YALEU/DCS/RR-397, june 1985.

3 Y. SAAD, *Chebychev Acceleration Techniques for Solving Non-symmetric Eigenvalue Problems*, Mathematics of Computation, vol. 42, pp. 567-588, April, 1984.

4 N KILL, M. GERADIN, *Analyse Dynamique des Machines Tournantes : Principes et Applications*, LTAS VF-50 Report, mars, 1984.

5. A. CARDONA, *An Integrated Approach to Mechanism Analysis*, Ph.D Thesis, University of Liège, 1989.

6. M. GERADIN, *Aspects Numériques et Logiciels de l'Approche Eléments Finis a l'Analyse Dynamique des Systèmes Articulés Flexibles*, Colloque National en Calcul des Structures, May 11-14 1993, Giens, France.

7 M. GERADIN, N. KILL, *Eigenvalue Algorithms for Stability and Critical Speeds Analysis of Rotating Systems*, 6th. International Modal Analysis Conference, February 1-4, 1988.

8. M. GERADIN, D. RIXEN, *Théorie des Vibrations*, MASON editions, 1993.

9. D. PEDELABORDE, *Dimensionnement au Flambage d'un Ressort de Torsion*, Journées SAMCEF 88, November 8-9, 1988.

10 SAMTECH S.A., *MECANO and DYNAM : User Manuals*, release 4.2, 1991.

Free-Floating Closed-Chain Planar Robots: Kinematics, Path Planning and Control

Sunil Kumar Agrawal, Assistant Professor
R. Garimella and G. Desmier, Graduate Students

Department of Mechanical Engineering
Ohio University, Athens, OH 45701

Abstract

The study of free-floating manipulators is important for the success of robotics programs in space and in the design of innovative robot systems which can operate over a large workspace In order to study the fundamental theoretical and experimental issues encountered in space robotics, a closed-chain planar manipulator was built at Ohio University (OU) which floats on a flat table using air bearings. Due to the absence of external forces in the plane of the table and couples normal to this plane, the linear momentum in the plane and the angular momentum normal to this plane are conserved It is well known that the linear momentum equations are holonomic while the angular momentum equation is nonholonomic Due to this nonholonomic behavior, the path-planning schemes commonly used for fixed-base manipulators do not directly apply to free-floating manipulators In this paper, we present an algorithm for motion planning of planar free-floating manipulators based on the inverse position kinematics of the mechanism. It is demonstrated that the inverse position kinematics algorithms, commonly used for fixed-base manipulators, can be successfully applied to free-floating manipulators using an iterative search procedure to satisfy the nonholonomic angular momentum constraints This procedure result in paths identical to those predicted by inverse rate kinematics The inverse position kinematics algorithm is then used to avoid singularities during motion to result in successful paths The results of the simulation of this algorithm using parameter estimates of the OU free-floating robot are presented

1 Introduction

Satellite-mounted manipulators will play an important role in the construction of the space station and in the future missions for space inspection and repair Over the last decade, several research have been reported on manipulation systems which have free-floating base Longman et al [7] presented a way to control the end-effector trajectories accounting for base motions Umetani and Yoshida ([13], [15]) presented the notion of generalized Jacobians and suggested an algorithm for dual-arm coordination of free-floating manipulators Alexander and Cannon [5] reported experimental results on a free-floating manipulator Vafa [14] presented the useful concept of 'virtual manipulators' applied to open-chain floating manipulators Papadopoulos and Dubowsky ([9], [10]) studied control algorithms and dynamic singularities of free-floating manipulators Schneider and Cannon [11] presented a strategy for cooperative manipulation of floating robots Agrawal et al [2] reported studies conducted on singularities of the Jacobian mapping of free-floating open-chain manipulators They also reported research conducted on workspace boundaries of free-floating closed-chain manipulators, and algorithms for dynamic simulation of these manipulators ([1], [3]) Mukherjee and Nakamura [8] proposed path planning algorithms for space manipulators

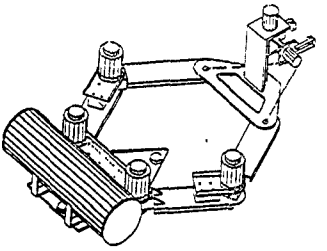


Figure 1 A 3-D solid model of a free-floating closed-chain manipulator built at Ohio University

From the above references, it is well established that free-floating manipulators are characterized by nonholonomic constraints Due to nonholonomic constraints, the conventional path-planning schemes applied to fixed-base manipulators are not directly applicable to free-floating manipulators The algorithms proposed in the literature for motion-planning of free-floating manipulators are either aimed at computing the joint torques using the dynamic equations of motion of the system ([10], [5], [15], [11]) or the joint rates using the rate kinematics of the system ([8], [4]) The approach presented in this paper differs from these two motion planning approaches as it uses the inverse position kinematics of the system for motion planning. The inverse position kinematics is a well accepted method for motion planning of fixed base manipulators but it has not been explored for free-floating manipulators due to the presence of non integrable angular momentum constraints In this paper, we demonstrate that using an iterative search procedure to satisfy the nonholonomic angular momentum constraint equation, the inverse position kinematics can be used for motion planning of nonholonomic systems The results of this algorithm using parameter estimates of the OU free-floating robot are presented

The paper is organized in the following way: A description of the OU free-floating closed-chain planar manipulator and its mathematical model is presented in Section 2 The position kinematic equations of the robot are developed in Section 3 The rate kinematic equations are developed in Section 4. The algorithm for motion planning using inverse position kinematics is presented in Section 5 These are followed by conclusions of this study

2 Kinematic Model of the Floating Manipulator

The free-floating planar manipulator, designed at OU, is a six revolute jointed closed chain mechanism A 3-D solid model of the robot is shown in Figure 1 The base of the robot floats on a flat granite surface using four air bearings The air is provided by a compressor mounted on the base The end-effector of this robot connects to the base via two series chains, each with three revolute joints The robot has 3 degrees of freedom and is driven by three D C servomotors[1] All six joints are instrumented with incremental optical encoders The robot is controlled by a PC 486 machine equipped with motion control boards An overhead X-Y position sensor (not shown in the figure) monitors the position of a moving point on the robot relative to the table

[1] A fourth dummy motor is shown in the solid model to counterbalance the mass on the two chains

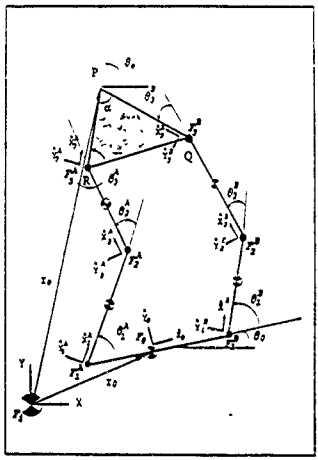


Figure 2 A mathematical model of the free-floating closed-chain planar manipulator

A mathematical model of this mechanism along with the link coordinate frames is shown in Figure 2 A coordinate frame $\mathcal{F}$ is fixed inertially at the center of mass of the mechanism A coordinate frame $\mathcal{F}_0$ is attached to the base link (link 0) of the robot at its center of mass The mass of this link is m_0 and the centroidal moment of inertia is I_0 A coordinate frame $\mathcal{F}_i^j$ is fixed to every link i of the chains $j \in A, B$ at the joint where it connects to the link $i-1$ The X and Y coordinate axes of the frame $\mathcal{F}_i^j$ are x_i^j and y_i^j respectively The relative joint displacement between two successive frames $\mathcal{F}_{i-1}^j$ and $\mathcal{F}_i^j$ is defined as θ_i^j The origins of $\mathcal{F}_1^A$ and $\mathcal{F}_1^B$ are located respectively at l_0^A and l_0^B from the origin of $\mathcal{F}_0$ along the $\hat{x}_0$ axis The length of link i in chain j is l_i^j The center of mass of a link i, $i = 1, 2$, of chain j is located at $l_{ci}^j \hat{x}_i^j$ relative to $\mathcal{F}_i^j$ The end-effector reference point P is measured in $\mathcal{F}$ by the vector $\mathbf{r}_e$. The point P is assumed to be the centroid of the end-effector link The mass of this link is m_e and the centroidal moment of inertia is I_e. The mass of a link i, $i = 1, 2$ of chain j is m_i^j and its centroidal moment of inertia is I_i^j

The parameters of the OU free-floating robot computed from a computer solid model are listed in Table 1 The free-floating manipulator is described by nine variables the three variables x_0, y_0, θ_0 describe the position and orientation of the base link in the inertial reference frame $\mathcal{F}$, the variables θ_1^A, θ_2^A, and θ_3^A describe the relative orientation of the links of chain A, and θ_1^B, θ_2^B, and θ_3^B are the relative orientation of the links of chain B The end-effector position is given by (x_e, y_e) and the orientation of the line QP in the fixed frame by the angle θ_e These twelve variables are the variables of interest for the manipulator

These twelve variables are subject to nine constraint equations The position vectors $\mathbf{r}_e$ from the

Base Link:
$l_0^A = 6.00$ in $l_0^B = 6.00$ in $m_0 = 56.00$ lb $I_0 = 1159.00$ lb in^2

Link 1^A and Link 1^B
$l_1^A = l_1^B = 12.00$ in $l_{c_1}^A = l_{c_1}^B = 8.30$ in $m_1^A = m_1^B = 4.75$ lb
$I_1^A = I_1^B = 8.07$ lb in^2

Link 2^A and Link 2^B
$l_2^A = l_2^B = 12.00$ in $l_{c_2}^A = l_{c_2}^B = 5.45$ in $m_2^A = m_2^B = 1.66$ lb
$I_2^A = I_2^B = 0.66$ lb in^2

End-effector
$l_3^A = l_3^B = 12.00$ in $l_{c_3}^A = l_{c_3}^B = 12.00$ in $m_e = 2.00$ lb
$I_e = 12.03$ lb in^2

End-effector cone angle.
$\alpha = 60°$

Table 1 The parameters of the OU free-floating robot computed from a computer solid model

origin of $\mathcal{F}$ to the reference point on the end-effector via chains A and B are

$$\begin{aligned} \mathbf{r}_e &= \mathbf{r}_0 + l_0^A \hat{\mathbf{x}}_0 + l_1^A \hat{\mathbf{x}}_1^A + l_2^A \hat{\mathbf{x}}_2^A + l_3^A \hat{\mathbf{x}}_3^A \\ &= \mathbf{r}_0 + l_0^B \hat{\mathbf{x}}_0 + l_1^B \hat{\mathbf{x}}_1^B + l_2^B \hat{\mathbf{x}}_2^B + l_3^B \hat{\mathbf{x}}_3^B \end{aligned} \tag{1}$$

where $\mathbf{r}_0$ is a vector from the origin of the inertially fixed frame to the origin of the base link This results in four scalar equations on the twelve variables The angle closure via the two chains results in

$$\begin{aligned} \theta_e &= \theta_0 + \theta_1^A + \theta_2^A + \theta_3^A \\ &= \theta_0 + \theta_1^B + \theta_2^B + \theta_3^B - \alpha \end{aligned} \tag{2}$$

where α is the cone angle of the end-effector plate as shown in the figure The linear momentum $\mathbf{P}$ of the mechanism is

$$\mathbf{P} = m_0 \dot{\mathbf{r}}_0 + m_1^A \dot{\mathbf{r}}_{c_1}^A + m_2^A \dot{\mathbf{r}}_{c_2}^A + m_e \dot{\mathbf{r}}_e + m_1^B \dot{\mathbf{r}}_{c_1}^B + m_2^B \dot{\mathbf{r}}_{c_2}^B \tag{3}$$

where $\mathbf{r}_0$, $\mathbf{r}_e$, $\mathbf{r}_{c_i}^j$ are respectively the velocity of the center of mass of the links In the absence of external forces in the plane, $\mathbf{P}$ is constant Assuming that the free-floating manipulator is released from rest, without loss of generality, $\mathbf{P} = 0$ The linear momentum equation (3) is integrable On integrating this equation and with the choice of $\mathcal{F}$ at the center of mass of the mechanism, this integral reduces to

$$m_0 \mathbf{r}_0 + m_1^A \mathbf{r}_{c_1}^A + m_2^A \mathbf{r}_{c_2}^A + m_e \mathbf{r}_e + m_1^B \mathbf{r}_{c_1}^B + m_2^B \mathbf{r}_{c_2}^B = 0 \tag{4}$$

where $\mathbf{r}_{c_i}^j$ is a vector from the origin of $\mathcal{F}$ to the mass center of the link i of chain j One interpretation of this equation is that the center of mass of the mechanism remains at the origin of $\mathcal{F}$ during the entire motion of the mechanism The position vectors $\mathbf{r}_{c_i}^j$ satisfy the following equation

$$\mathbf{r}_{c_i}^j = \mathbf{r}_0 + \sum_{k=0}^{i-1} (l_k^j \hat{\mathbf{x}}_k^j) + l_{c_i}^j \hat{\mathbf{x}}_i^j \qquad i = 1, \ldots, 3 \qquad j = A, B \tag{5}$$

The angular momentum H about the center of mass of the mechanism is computed by summing up the angular momentum contributions from the individual links. This angular momentum is

$$\begin{aligned}\mathbf{H} &= m_0(\mathbf{r}_0 \times \dot{\mathbf{r}}_0) + m_e(\mathbf{r}_e \times \dot{\mathbf{r}}_e) \\ &+ \sum_{i=1}^{2}[m_i^A(\mathbf{r}_{ci}^A \times \dot{\mathbf{r}}_{ci}^A + m_i^B(\mathbf{r}_{ci}^B \times \dot{\mathbf{r}}_{ci}^B] \\ &+ [I_0\dot{\theta}_0 + I_e\dot{\theta}_e + \sum_{i=1}^{2}\{I_i^A(\dot{\theta}_0 + \sum_{j=1}^{i}\dot{\theta}_j^A) + \\ &I_i^B(\dot{\theta}_0 + \sum_{j=1}^{i}\dot{\theta}_j^B)\}]\hat{z} \end{aligned} \tag{6}$$

Due to the planar structure of the robot, H is along a direction normal to the plane of motion of the robot. In the absence of external forces in the plane and couples normal to the plane, H is a constant. Since the system starts from rest, $\mathbf{H} = 0$. This angular momentum equation is not integrable and is a nonholonomic equation.

In summary, the free-floating manipulator is described by 12 variables. These 12 variables satisfy eight position constraint equations, four position loop closure eqns. (1), two angle loop closure eqns. (2), two center of mass constraint eqns. (4), and one non-integrable rate constraint eq. (6).

3 Inverse Position Kinematics

The inverse position kinematics problem is to determine the joint angles at a given end-effector position and orientation. For given end-effector position (x_e, y_e) and orientation θ_e, the eight position constraints outlined in the last section are not enough to give a unique solution as the angular momentum equation can not be used in the solution of the inverse position kinematics. One more variable, which we choose as the base orientation θ_0, needs to be specified to solve for the angles uniquely. As quite evident, any arbitrary value of θ_0 can not be chosen for the inverse solution at a point but it must be such that the solutions at two successive points satisfy the angular momentum equation. With this motivation in mind, the path-planning problem has been broken into two steps: (a) obtain the inverse solutions at a given (x_e, y_e, θ_e) using only the 8 position constraint equations with the base angle θ_0 treated as a free variable, (b) use an iterative search procedure to determine θ_0 such that the joint angles satisfy the difference form of the angular momentum constraint equation. The part (a) of this procedure is described in this section while the part (b) is described in Section 5. In this section, we also show that the inverse kinematic solutions of free-floating robots have features very different from those of the inverse solutions of structurally similar fixed-base manipulators.

On equating the expressions for $\mathbf{r}_e - \mathbf{r}_0$ from eq. (1), we obtain

$$a_{11}\mathbf{x}_0 + a_{12}\mathbf{x}_1^A + a_{13}\mathbf{x}_2^A + a_{14}\mathbf{x}_1^B + a_{15}\mathbf{x}_2^B = \mathbf{p}_1 \tag{7}$$

Substituting eq. (5) into (4), we obtain

$$A_1\mathbf{r}_0 + B_1\mathbf{x}_0 + C_1\mathbf{x}_1^A + D_1\mathbf{x}_2^A + E_1\mathbf{x}_3^A + F_1\mathbf{x}_1^B + G_1\mathbf{x}_2^B = 0 \tag{8}$$

On solving $\mathbf{r}_0$ from eq. (1) and substituting in the above equation, we obtain

$$a_{21}\mathbf{x}_0 + a_{22}\mathbf{x}_1^A + a_{23}\mathbf{x}_2^A + a_{24}\mathbf{x}_1^B + a_{25}\mathbf{x}_2^B = \mathbf{p}_2 \tag{9}$$

All coefficients and vectors appearing in the above three equations are listed in Appendix I.

The eqs. (7) and (9) involve the coefficients a_{ij} and the vectors $\mathbf{p}_1$, $\mathbf{p}_2$ which are known at a given $\mathbf{r}_e$. When resolved along the coordinate directions of $\mathcal{F}$, these two vector equations result in four

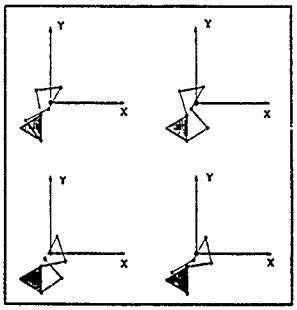


Figure 3: The inverse solutions for an end-effector position $(-15\ 0, -12\ 0)$, orientation angle $\theta_e = 150°$ and base angle $\theta_0 = 60°$.

scalar equations which involve 'cos' and 'sin' of the angles θ_0, θ_{01}^A, θ_{02}^A, θ_{01}^B, and θ_{02}^B. A notation such as θ_{02}^A is $\theta_0 + \sum_{j=1}^{2} \theta_j^A$. The four scalar equations must be simultaneously solved for the four joint angles. On treating θ_0 as a free variable and using the procedure described in Appendix 1, the four equations can be reduced to one 8th degree polynomial equation in the variable z_1

$$a_0 + a_1 z_1 + a_2 z_1^2 + a_3 z_1^3 + a_4 z_1^4 + a_5 z_1^5 + a_6 z_1^6 + a_7 z_1^7 + a_8 z_1^8 = 0 \tag{10}$$

where $z_1 = tan(\frac{\theta_{01}^B}{2})$.

The zeros of this polynomial are the solutions of z_1 which satisfy equations (7) and (9). Upto 8 roots of z_1 (or θ_{01}^B) are theoretically possible. On substituting one of the possible eight values of z_1 in equations (30) and (31), we obtain two quadratic equations in z_2. These two quadratics result in one common solution. Hence, a maximum of 8 solutions of $(\theta_{01}^B, \theta_{02}^B)$ are possible. For each feasible $(\theta_{01}^B, \theta_{02}^B)$, there is a unique solution for θ_{01}^A and θ_{02}^A from equation (26) and (27). From this procedure, theoretically, for each end-effector position/orientation and a specified orientation of the base link, upto 8 solutions of the joint angles are possible. The details of this algorithm are available in [6].

Figure 3 shows a set of four solutions for an end-effector position and orientation with the parameters of the OU free-floating robot. It may be instructive to contrast these inverse solutions with those of a structurally similar fixed base manipulator. It is well known that for the latter case, there are four solutions and the solutions are two pairs of mirror images. For the free-floating case, we see from the figure that there are no mirror images in the four solutions.

4 Inverse Rate Kinematics

Eventhough the focus of this paper is to provide a motion planning algorithm using inverse position kinematics, it is important to develop the forward and inverse rate kinematics relationships between the end-effector and the joint variables. The inverse rate kinematics, on direct integration, results in

acceptable motion plans for the system These motion plans obatined via direct integration provide an easy way to compare and evaluate the effectiveness of the motion planning algorithm using inverse position kinematics In this section, we will describe the rate relations in two steps. (1) the constraints on the nine rates describing the mechanism x_0, y_0, θ_0, θ_1^A, θ_2^A, θ_3^A, θ_1^B, θ_2^B, θ_3^B and (2) the mapping between the end-effector rates x_e, y_e, θ_e and the joint rates.

4.1 Joint Rate Constraints

On differentiating the X, Y position loop closure equations and the angle loop closure equation, we obtain three constraints on the nine rate variables. On collecting the terms of the X, Y linear momentum constraint and the angular momentum constraint, we obtain another three constraints on these rate variables. These six constraint equations can be written in a form $Kq = 0$, where K is a (6×9) matrix:

$$\begin{bmatrix} 0 & 0 & 0 & K_{14} & K_{15} & K_{16} & K_{17} & K_{18} & K_{19} \\ 0 & 0 & 0 & K_{24} & K_{25} & K_{26} & K_{27} & K_{28} & K_{29} \\ 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 \\ K_{41} & 0 & K_{43} & K_{44} & K_{45} & 0 & K_{47} & K_{48} & K_{49} \\ 0 & K_{52} & K_{53} & K_{54} & K_{55} & 0 & K_{57} & K_{58} & K_{59} \\ 0 & 0 & K_{63} & K_{64} & K_{65} & 0 & K_{67} & K_{68} & K_{69} \end{bmatrix} \tag{11}$$

and the vector q is $(x_0, y_0, \theta_0, \theta_1^A, \theta_2^A, \theta_3^A, \theta_1^B, \theta_2^B, \theta_3^B)^T$ The constants K_{42}, K_{51}, K_{61}, and K_{62} are identically zero. The remaining coefficients are listed in Appendix 2. Using the above matrix, six rates can be solved in terms of the remaining three. These three independent rate variables play the role of the active or the controlled joints. Defining $\mathbf{q}_f$ as the vector of the three active joint rates and $\mathbf{q}_r$ as the vector of the remaining six joint rates, the equation $Kq = 0$ can be rewritten as:

$$K_f \mathbf{q}_f + K_r \mathbf{q}_r = 0. \tag{12}$$

where K_f and K_r are partitions of K K_f is a (6×3) matrix and K_r is a (6×6) matrix $\mathbf{q}_f$ is a (3×1) vector and $\mathbf{q}_r$ is a (6×1) vector Using eq (12), $\mathbf{q}_r$ can be solved in terms of $\mathbf{q}_f$

$$\mathbf{q}_r = -K_r^{-1} K_f \mathbf{q}_f \tag{13}$$

If the elements of the vector q are permuted according to the choice of the active joints, such that $\mathbf{q} = P\mathbf{q}'$, using eq. (13), the vector $\mathbf{q}'$ can be written in terms of $\mathbf{q}_f$ as

$$\mathbf{q}' = \begin{bmatrix} \mathbf{q}_r \\ \mathbf{q}_f \end{bmatrix} = C\mathbf{q}_f \tag{14}$$

where C is a (9×3) matrix formed by augmenting a (3×3) identity matrix E_3 with the (6×3) matrix $-K_r^{-1}K_f$ This matrix C is

$$C = \begin{bmatrix} -K_r^{-1} K_f \\ E_3 \end{bmatrix} \tag{15}$$

4.2 End-Effector Rate Constraints

The end-effector rates x_e, y_e, and θ_e can be related to the joint rates in the following way. The position closure eq (1) and the angle closure eq. (2) obtained via chain A are differentiated and collected These end-effector rates as a function of the joint rates are

$$\mathbf{X}_e = P\mathbf{q} \tag{16}$$

where $\mathbf{X}_e = (x_e, y_e, \theta_e)^T$, P is a (3×9) matrix The structure of P is as follows

$$\begin{bmatrix} 1 & 0 & P_{13} & P_{14} & P_{15} & P_{16} & 0 & 0 & 0 \\ 0 & 1 & P_{23} & P_{24} & P_{25} & P_{26} & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

The terms appearing in the above equation are defined in eq (42). Using eq (16)and (14), the mapping between the end-effector rates and the active joint rates is

$$\dot{\mathbf{X}}_e = J\mathbf{q}_a \tag{18}$$

where $J = PPC$. J is the well-known Jacobian mapping between the end-effector rates and the active joint rates. In order to determine the joint rates for specified end-effector rates, the following 9 first-order nonlinear differential equations must be integrated·

$$\mathbf{q} = PCJ^{-1}\mathbf{X}_e \tag{19}$$

For successful motion planning of the robot, the following two conditions must be guaranteed. J exists and is invertible. J exists only when the matrix K_r is invertible. Hence, we will find the conditions for the invertibility of K_r.

4.3 Invertibility of K_r

The structure of the matrix K_r is dependent on the choice of active joints for the mechanism In order to give a geometric interpretation of the invertibility of the matrix K_r, it will be best to choose a set of active joints and partition the matrix K_r. Since the coordinates x_0, y_0, and θ_0 can not be directly controlled, they are not good choices for the active joints. For the sake of of discussion, we choose the active joint vector as $\mathbf{q}_a = (\theta_1^A, \theta_2^A, \theta_3^A)^T$, the matrix K_r becomes:

$$K_r = \begin{bmatrix} 0 & 0 & 0 & K_{17} & K_{18} & K_{19} \\ 0 & 0 & 0 & K_{27} & K_{28} & K_{29} \\ 0 & 0 & 0 & -1 & -1 & -1 \\ K_{41} & 0 & K_{43} & K_{47} & K_{48} & K_{49} \\ 0 & K_{52} & K_{53} & K_{57} & K_{58} & K_{59} \\ 0 & 0 & K_{63} & K_{67} & K_{68} & K_{69} \end{bmatrix} \tag{20}$$

The above matrix has the following structure

$$K_r = \begin{bmatrix} O & Q \\ R & S \end{bmatrix} \tag{21}$$

where O, Q, R, and S are (3×3) matrices and all elements of O are zeros R is an upper triangular matrix. The inverse of the matrix K_r is

$$K_r^{-1} = \begin{bmatrix} -R^{-1}SQ^{-1} & R^{-1} \\ Q^{-1} & O \end{bmatrix} \tag{22}$$

From the above equation, we see that the inverse of K_r exists only when the inverse of the matrices Q and R exist Since R is an upper triangular matrix, the determinant of this matrix is zero only when one of its three diagonal terms K_{41}, K_{52}, K_{63} are zero On examining K_{41} and K_{52}, we find that they are the total mass of the system, hence, are never zeros Physically, K_{63} is the sum of the moments of inertia and is also nonzero Hence, the matrix R is never singular

The matrix Q is formed from the three loop closure equations These three equations for the free-floating mechanism are identical to the position and angle loop closure equations of a structurally

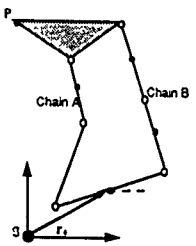


Figure 4 A configuration in which the joints 1, 2, and 3 of chain B are in a line At this configuration, K_r is singular if θ_1^A, θ_2^A, and θ_3^A are chosen as the active joints.

similar fixed-base mechanism. Hence, the configurations where Q is singular is identical to the singular configurations of a structurally similar fixed-base mechanism. From the work in literature (e g , [12]), the matrix Q is not invertible whenever a line passes through the three joints of chain B, i e., the chain B is fully extended outwards or is fully folded inwards. A configuration where K_r is singular is shown in Figure 4. Similarly, it can be proved that for a set of any three active joints among the joints of chain A or chain B, the condition of singularity of the corresponding K_r matrix is when the three unactuated joints are aligned, i e., a line passes through the three joints

4.4 Singularity of J

The singularities of the J matrix have also been termed as the 'dynamic singularities' in the literature [9]. Unlike the K_r matrix, it is not possible to provide a geometric interpretation of the configurations where the J matrix is singular.

5 Path Planning

One method to generate the motion plans is by direct integration of the rate eqs (19) by choosing a desired function $\mathbf{X}_e(t)$ In general, this method will work for short paths and will fail at the configurations where K_r or J become singular.

The inverse position kinematics algorithm can be also used to plan paths between an initial and a final position/orientation of the end-effector The essence of the algorithm for motion planning paths between two points is described in Table 2 A feasible assembly configuration is assumed at the starting point of the path The path is divided into a number of intermediate points At a step $i+1$ of the algorithm, the current value of the base orientation $(\theta_0)_i$ is used to deter ine the search interval for $(\theta_0)_{i+1}$. The search interval for $(\theta_0)_{i+1}$ is discretized and correspondi o a value $(\theta_0)_{i+1}$, the inverse kinematics procedure is used to compute the joint angles. The multiple solutions obtained by the inverse kinematics procedure are sorted to determine the best solution near to the joint angles at the step i With this best solution, the following difference form of the angular momentum equation is evaluated.

$$K_{61}\Delta x_0 + K_{62}\Delta y_0 + K_{63}\Delta\theta_0 + K_{64}\Delta\theta_1^A + K_{65}\Delta\theta_2^A + K_{66}\Delta\theta_3^A + K_{67}\Delta\theta_1^B$$

```
Discretize end-effector path
Obtain Start configuration
For i = 1 to number of points
    {
    For (θ_6)_{i-1} = (θ_6)_i - δ to (θ_6)_i + δ
        {
        Solve inverse kinematics
        If there is a solution then
            {
            Determine minimum norm solution q_{i+1}
            Evaluate angular momentum equation
            If angular momentum equation has changed sign then
                {
                q_{i+1} is the approximate solution
                }
            }
        }
    }
```

Table 2: The algorithm to obtain motion plans using inverse kinematics.

$$+K_{68}\,\Delta\,\theta_2^B + K_{69}\,\Delta\,\theta_3^B \tag{23}$$

where the symbol Δ is used for the difference of a joint variable computed between the steps $i+1$ and i. A feasible joint solution satisfies eq. (23) within a specified tolerance It is important to point out that the motion plans obtained using this algorithm are almost identical to that obtained using direct integration of the inverse rate kinematics equation (19). Figure 5 shows a successful path between two points in the workspace. The same path is also planned using direct integration of the rate kinematics. The base orientation angle θ_0, the angles θ_1^A, θ_2^A, and θ_3^A for this path using the direct integration of the rate kinematics and the inverse position kinematics are presented in the same figure for easy comparison

It must be pointed out, however, that the use of the inverse position kinematics algorithm does not alleviate the problem of singularities encountered in the direct integration of the rate kinematics equations. In fact, one can demonstrate that the inverse position kinematics and the inverse rate kinematics algorithms break at exactly the same point when encountering a singularity In case of rate kinematics, due to ill-conditioning of the J matrix, the algorithm breaks down. In case of inverse position kinematics algorithms, the search procedure does not find a base angle $(\theta_0)_{i+1}$ close to $(\theta_0)_i$ which satisfies the difference form of the angular momentum constraint equation.

The inverse position kinematics algorithm, however, can be easily modified to avoid singularities where it can not find a set of joint angles that satisfy the difference form of the angular momentum constraint equation. These singularities are avoided by taking a deviation in the Cartesian path in order to satisfy the angular momentum constraint equation. Figure 6 shows the joint angles for a straight line path in the Cartesian coordinates using the three algorithms proposed in this paper, i e , (a) direct integration of the rate kinematics, (b) inverse position kinematics without excursions at the singularities, and (c) inverse position kinematics allowing for excursions of the end-effector point near a singularity This figure shows the Cartesian path, the base orientation angle, and the joint angles θ_1^A, θ_2^A From the figure, we observe that the integration of the rate kinematics breaks down in the middle of the path The inverse position kinamtics algorithm goes through jumps in the joint angles at the positions of the singularity. The exptended inverse kinematics algorithm which allows excursions in the end-effector path is smooth at the singularities and successfully reaches the goal

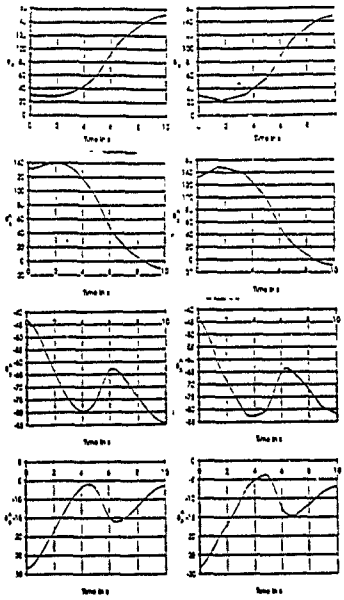


Figure 5 A successful path plan using (a) direct integration of the rate kinematics equation, (b) inverse position kinematics algorithm

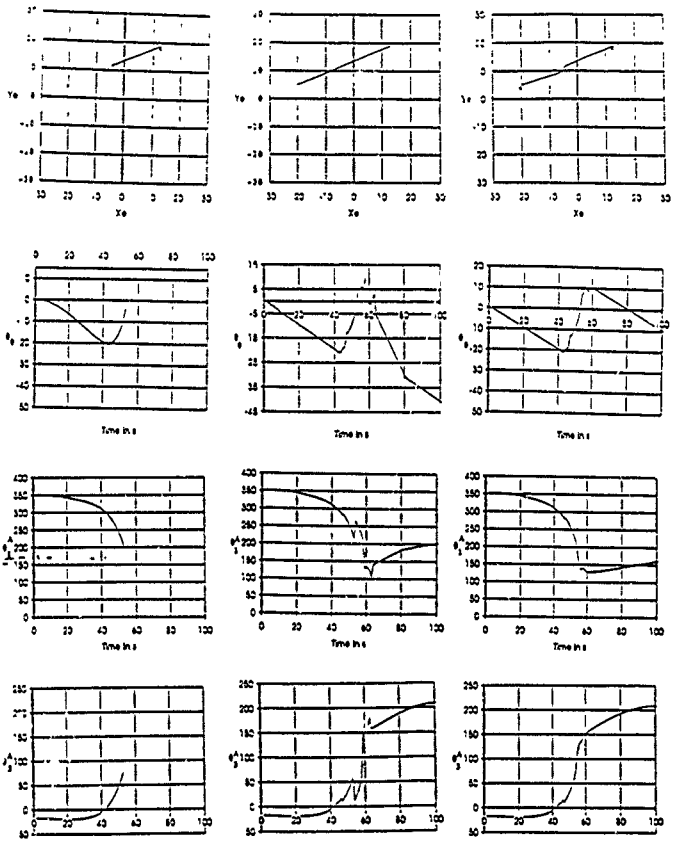


Figure 6: The motion plan for a straight line path in the Cartesian coordinates using the three algorithms proposed in this paper, i.e., (a) direct integration of the rate kinematics, (b) inverse position kinematics without excursions at the singularities, and (c) inverse position kinematics allowing for excursions of the end-effector point near a singularity

6 Conclusions

In this paper, we presented an algorithm for motion-planning of free-floating robots using inverse position kinematics Eventhough free-floating manipulators are characterized by nonholonomic constraints, we showed that the inverse position kinematics coupled with an iterative search procedure to satisfy the nonholonomic constraints results in motion plans predicted by direct integration of the rate equations The problem of singularities which are encountered during the integration of the rate kinematic equations are still faced in the inverse position kinematics algorithm. However, it is easy to take excursions in the path using the inverse position kinematics algorithm to avoid the singular configurations and reach the goal. The results obtained using the inverse kinematics algorithm were compared with the direct integration of the rate kinematics equations to highlight the similarities and differences between the two approaches The authors feel that this method of motion planning which was applied to free-floating manipulators can be successfully applied to other classes of nonholonomic systems

7 Acknowledgments

The support of the Dynamic Systems and Control Division of the NSF under the Research Initiation Program and Baker Fund Awards for the first author are gratefully acknowledged.

8 Appendix 1

8.1 Derivation of 8th degree polynomial

All coefficients appearing in this section are listed in Section 8.2 The equations (7) and (9) when resolved along the coordinate directions of $\mathcal{F}$ have the following form·

$$\begin{bmatrix} \alpha_{12} & \alpha_{13} \\ \alpha_{22} & \alpha_{23} \end{bmatrix} \begin{bmatrix} c^A_{01} \\ c^A_{02} \end{bmatrix} + \begin{bmatrix} \alpha_{14} & \alpha_{15} \\ \alpha_{24} & \alpha_{25} \end{bmatrix} \begin{bmatrix} c^B_{01} \\ c^B_{02} \end{bmatrix} = \begin{bmatrix} p'_{1x} \\ p'_{2x} \end{bmatrix} \tag{24}$$

and

$$\begin{bmatrix} \alpha_{12} & \alpha_{13} \\ \alpha_{22} & \alpha_{23} \end{bmatrix} \begin{bmatrix} s^A_{01} \\ s^A_{02} \end{bmatrix} + \begin{bmatrix} \alpha_{14} & \alpha_{15} \\ \alpha_{24} & \alpha_{25} \end{bmatrix} \begin{bmatrix} s^B_{01} \\ s^B_{02} \end{bmatrix} = \begin{bmatrix} p'_{1y} \\ p'_{2y} \end{bmatrix} \tag{25}$$

The eqs (24) and (25) are used to solve for the 'sin' and 'cos' of the angles θ^A_{01} and θ^A_{02} in terms of joint angles θ^B_{01} and θ^B_{02}

$$\begin{bmatrix} c^A_{01} \\ c^A_{02} \end{bmatrix} = \begin{bmatrix} \beta_{11}c^B_{01} + \beta_{12}c^B_{02} + \beta_{13} \\ \beta_{21}c^B_{01} + \beta_{22}c^B_{02} + \beta_{23} \end{bmatrix} \tag{26}$$

and

$$\begin{bmatrix} s^A_{01} \\ s^A_{02} \end{bmatrix} = \begin{bmatrix} \beta_{11}s^B_{01} + \beta_{12}s^B_{02} + \beta_{13}' \\ \beta_{21}s^B_{01} + \beta_{22}s^B_{02} + \beta_{23}' \end{bmatrix} \tag{27}$$

The angles θ^A_{01} and θ^A_{02} can be eliminated by squaring and adding the component equations of (26) and (27) The resulting equations have the following form:

$$\begin{aligned} &\gamma_{11}(c^B_{01}c^B_{02} + s^B_{01}s^B_{02}) + \gamma_{12}c^B_{01} \\ &\quad + \gamma_{13}s^B_{01} + \gamma_{14}c^B_{02} + \gamma_{15}s^B_{02} + \gamma_{16} = 0 \end{aligned} \tag{28}$$

$$\begin{aligned} &\gamma_{21}(c^B_{01}c^B_{02} + s^B_{01}s^B_{02}) + \gamma_{22}c^B_{01} \\ &\quad + \gamma_{23}s^B_{01} + \gamma_{24}c^B_{02} + \gamma_{25}s^B_{02} + \gamma_{26} = 0 \end{aligned} \tag{29}$$

On substitution, $c_{01}^B = \frac{1-x_1^2}{1+x_1^2}$, $s_{01}^B = \frac{2x_1}{1+x_1^2}$, $c_{02}^B = \frac{1-x_2^2}{1+x_2^2}$, and $s_{02}^B = \frac{2x_2}{1+x_2^2}$, where $x_1 = tan(\frac{\theta_{01}^B}{2})$ and $x_2 = tan(\frac{\theta_{02}^B}{2})$, the above equations can be written as polynomials in x_1 and x_2

$$\begin{aligned}&\psi_{11}x_1^2x_2^2 + \psi_{12}x_1x_2^2 + \psi_{13}x_1^2x_2 + \psi_{14}x_1^2\\&+\psi_{15}x_2^2 + \psi_{16}x_1x_2 + \psi_{17}x_1 + \psi_{18}x_2 + \psi_{19} = 0 \qquad (30)\\&\psi_{21}x_1^2x_2^2 + \psi_{22}x_1x_2^2 + \psi_{23}x_1^2x_2 + \psi_{24}x_1^2\\&+\psi_{25}x_2^2 + \psi_{26}x_1x_2 + \psi_{27}x_1 + \psi_{28}x_2 + \psi_{29} = 0 \qquad (31)\end{aligned}$$

On collecting the coefficients of x_2^2, x_2, and 1, these two equatons can be written in the following form:

$$\begin{bmatrix} \psi_{11}x_1^2 + \psi_{12}x_1 + \psi_{15} & \psi_{13}x_1^2 + \psi_{16}x_1 + \psi_{18} \\ \psi_{21}x_1^2 + \psi_{22}x_1 + \psi_{25} & \psi_{23}x_1^2 + \psi_{26}x_1 + \psi_{28} \\ \psi_{14}x_1^2 + \psi_{17}x_1 + \psi_{19} & \\ \psi_{24}x_1^2 + \psi_{27}x_1 + \psi_{29} & \end{bmatrix} \begin{bmatrix} x_2^2 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (32)$$

The set of the above two polynomial equations must be simultaneously solved for the variables x_1 and x_2. The method of 'dialytic elimination' is adopted to obtain the solutions. Assuming $x_2 \neq 0$, the two equations are multiplied by x_2 to form additional two equations The above two equations of (32) and the two additional equations generated when by multiplied by x_2 can now be written in a matrix form:

$$\begin{bmatrix} \pi_{11} & \pi_{12} & \pi_{13} & 0 \\ \pi_{21} & \pi_{22} & \pi_{23} & 0 \\ 0 & \pi_{11} & \pi_{12} & \pi_{13} \\ 0 & \pi_{21} & \pi_{22} & \pi_{23} \end{bmatrix} \begin{bmatrix} x_2^3 \\ x_2^2 \\ x_2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (33)$$

where π_{ij} are coefficients of the matrix in equation (32) For this set of four equations to have a common solution, the determinant of the (4 x 4) matrix in eq. (33) with coefficients π_{ij} must vanish Mathematically,

$$\begin{aligned}&(\pi_{11}\pi_{22} - \pi_{12}\pi_{21})(\pi_{12}\pi_{23} - \pi_{22}\pi_{13})\\&-(\pi_{11}\pi_{23} - \pi_{21}\pi_{13})^2 = 0 \qquad (34)\end{aligned}$$

Making these substitutions,

$$\begin{aligned}\pi_{11}\pi_{22} - \pi_{12}\pi_{21} &= \sum_{i=0}^{4} a_{1i}x_1^i\\ \pi_{12}\pi_{23} - \pi_{22}\pi_{13} &= \sum_{i=0}^{4} a_{2i}x_1^i\\ \pi_{11}\pi_{23} - \pi_{13}\pi_{21} &= \sum_{i=0}^{4} a_{3i}x_1^i \qquad (35)\end{aligned}$$

the resulting determinant is a 8th degree polynomial in x_1 and has the following form

$$\begin{aligned}&a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 + a_4x_1^4 + a_5x_1^5\\&+a_6x_1^6 + a_7x_1^7 + a_8x_1^8 = 0 \qquad (36)\end{aligned}$$

8.2 Coefficients of the inverse position kinematics

In equation (8), the coefficients are

$$\begin{aligned}
A_1 &= m_0 + m_1^A + m_2^A + m_1^B + m_2^B + m_3^A \\
B_1 &= l_0^A(m_1^A + m_2^A + m_3^A) + l_0^B(m_1^B + m_2^B) \\
C_1 &= m_1^A l_{x1}^A + l_1^A(m_2^A + m_3^A) \\
D_1 &= m_2^A l_{x2}^A + m_3^A l_2^A \\
E_1 &= m_3^A l_3^A \\
F_1 &= m_1^B l_{x1}^B + m_2^B l_2^B \\
G_1 &= m_2^B l_{x2}^B
\end{aligned} \tag{37}$$

The coefficients appearing in equations (7) and (9) are

$$\begin{aligned}
&\alpha_{11} = l_0^A - l_0^B, \quad \alpha_{12} = l_1^A, \quad \alpha_{13} = l_2^A \\
&\alpha_{14} = -l_1^B, \quad \alpha_{15} = -l_2^B, \quad \mathrm{p}_1 = l_3^B x_3^B - l_3^A x_3^A \\
&\alpha_{21} = B_1 - A_1 l_0^B, \quad \alpha_{22} = C_1, \quad \alpha_{23} = D_1 \\
&\alpha_{24} = F_1 - A_1 l_1^B, \quad \alpha_{25} = G_1 - A_1 l_2^B \\
&\mathrm{p}_2 = -A_1 x_e - E_1 \dot{x}_3^A + A_1 l_3^B \dot{x}_3^B
\end{aligned} \tag{38}$$

In equations (24), and (25), the X and Y components of p_1' and p_2' are

$$\begin{aligned}
\mathrm{p}_{1x}' &= (l_0^B - l_0^A)c_0 + l_3^B c_{03}^B - l_3^A c_{03}^A \\
\mathrm{p}_{1y}' &= (l_0^B - l_0^A)s_0 + l_3^B s_{03}^B - l_3^A s_{03}^A \\
\mathrm{p}_{2x}' &= -A_1 x_e - (B_1 - A_1 l_0^B)c_0 - E_1 c_{03}^A + A_1 l_3^B c_{03}^B \\
\mathrm{p}_{2y}' &= -A_1 y_e - (B_1 - A_1 l_0^B)s_0 - E_1 s_{03}^A + A_1 l_3^B s_{03}^B
\end{aligned} \tag{39}$$

In equation (26) and (27), the coefficients are

$$\begin{aligned}
&\beta_{11} = \frac{-\alpha_{14}\alpha_{23} + \alpha_{13}\alpha_{24}}{D}, \quad \beta_{12} = \frac{-\alpha_{15}\alpha_{23} + \alpha_{13}\alpha_{25}}{D} \\
&\beta_{21} = \frac{\alpha_{14}\alpha_{22} - \alpha_{12}\alpha_{24}}{D}, \quad \beta_{22} = \frac{\alpha_{15}\alpha_{22} - \alpha_{12}\alpha_{25}}{D} \\
&\beta_{13} = \frac{\alpha_{23}\mathrm{p}_{1x}' - \alpha_{13}\mathrm{p}_{2x}'}{D}, \quad \beta_{23} = \frac{-\alpha_{22}\mathrm{p}_{1x}' + \alpha_{12}\mathrm{p}_{2x}'}{D} \\
&\beta_{13}' = \frac{\alpha_{23}\mathrm{p}_{1y}' - \alpha_{13}\mathrm{p}_{2y}'}{D}, \quad \beta_{23}' = \frac{-\alpha_{22}\mathrm{p}_{1y}' + \alpha_{12}\mathrm{p}_{2y}'}{D}
\end{aligned} \tag{40}$$

where $D = \alpha_{12}\alpha_{23} - \alpha_{13}\alpha_{22}$ The coefficients in equation (28) and (29) are

$$\begin{aligned}
&\gamma_{11} = 2\beta_{11}\beta_{12}, \quad \gamma_{12} = 2\beta_{11}\beta_{13}, \\
&\gamma_{13} = 2\beta_{11}\beta_{13}', \quad \gamma_{14} = 2\beta_{12}\beta_{13}, \\
&\gamma_{15} = 2\beta_{12}\beta_{13}', \quad \gamma_{16} = \beta_{11}^2 + \beta_{12}^2 + \beta_{13}^2 + \beta_{13}'^2 - 1 \\
&\gamma_{21} = 2\beta_{21}\beta_{22}, \quad \gamma_{22} = 2\beta_{21}\beta_{23} \\
&\gamma_{23} = 2\beta_{21}\beta_{23}', \quad \gamma_{24} = 2\beta_{22}\beta_{23} \\
&\gamma_{25} = 2\beta_{22}\beta_{23}', \quad \gamma_{26} = \beta_{21}^2 + \beta_{22}^2 + \beta_{23}^2 + \beta_{23}'^2 - 1
\end{aligned} \tag{41}$$

In equation (30) and (31), the coefficients ψ_{ij} are

$$\begin{aligned}
&\psi_{11} = \gamma_{11} - \gamma_{12} - \gamma_{14} + \gamma_{16}, \quad \psi_{12} = 2\gamma_{13} \\
&\psi_{13} = 2\gamma_{15}, \quad \psi_{14} = -\gamma_{11} - \gamma_{12} + \gamma_{14} + \gamma_{16} \\
&\psi_{15} = -\gamma_{11} + \gamma_{12} - \gamma_{14} + \gamma_{16}, \quad \psi_{16} = 4\gamma_{11} \\
&\psi_{17} = 2\gamma_{13}, \quad \psi_{18} = 2\gamma_{15} \\
&\psi_{19} = \gamma_{11} + \gamma_{12} + \gamma_{14} + \gamma_{16} \\
&\psi_{21} = \gamma_{21} - \gamma_{22} - \gamma_{24} + \gamma_{26}, \quad \psi_{22} = 2\gamma_{23} \\
&\psi_{23} = 2\gamma_{25}, \quad \psi_{24} = -\gamma_{21} - \gamma_{22} + \gamma_{24} + \gamma_{26} \\
&\psi_{25} = -\gamma_{21} + \gamma_{22} - \gamma_{24} + \gamma_{26}, \quad \psi_{26} = 4\gamma_{21} \\
&\psi_{27} = 2\gamma_{23}, \quad \psi_{28} = 2\gamma_{25} \\
&\psi_{29} = \gamma_{21} + \gamma_{22} + \gamma_{24} + \gamma_{26}
\end{aligned} \tag{42}$$

In equation (35), the coefficients are

$$\begin{aligned}
a_{10} &= \psi_{15}\psi_{28} - \psi_{25}\psi_{18} \\
a_{11} &= \psi_{12}\psi_{28} + \psi_{15}\psi_{26} - \psi_{22}\psi_{18} - \psi_{25}\psi_{15} \\
a_{12} &= \psi_{11}\psi_{28} + \psi_{12}\psi_{26} + \psi_{15}\psi_{23} - \psi_{21}\psi_{18} \\
&\quad - \psi_{22}\psi_{15} - \psi_{25}\psi_{13} \\
a_{13} &= \psi_{11}\psi_{26} + \psi_{12}\psi_{23} - \psi_{21}\psi_{15} - \psi_{22}\psi_{13} \\
a_{14} &= \psi_{11}\psi_{23} - \psi_{21}\psi_{13}
\end{aligned} \tag{43}$$

The coefficients a_{2i} are

$$\begin{aligned}
a_{20} &= \psi_{18}\psi_{29} - \psi_{19}\psi_{28} \\
a_{21} &= \psi_{16}\psi_{29} + \psi_{18}\psi_{27} - \psi_{19}\psi_{26} - \psi_{17}\psi_{28} \\
a_{22} &= \psi_{13}\psi_{29} + \psi_{16}\psi_{27} + \psi_{18}\psi_{24} - \psi_{23}\psi_{19} \\
&\quad - \psi_{26}\psi_{17} - \psi_{28}\psi_{14} \\
a_{23} &= \psi_{13}\psi_{27} + \psi_{24}\psi_{16} - \psi_{14}\psi_{26} - \psi_{17}\psi_{23} \\
a_{24} &= \psi_{13}\psi_{24} - \psi_{14}\psi_{23}
\end{aligned} \tag{44}$$

and the coefficients a_{3i} are

$$\begin{aligned}
a_{30} &= \psi_{15}\psi_{29} - \psi_{19}\psi_{25} \\
a_{31} &= \psi_{12}\psi_{29} + \psi_{15}\psi_{27} - \psi_{19}\psi_{22} - \psi_{17}\psi_{25} \\
a_{32} &= \psi_{11}\psi_{29} + \psi_{15}\psi_{24} + \psi_{12}\psi_{27} - \psi_{21}\psi_{19} \\
&\quad - \psi_{25}\psi_{14} - \psi_{17}\psi_{22} \\
a_{33} &= \psi_{11}\psi_{27} + \psi_{12}\psi_{24} - \psi_{21}\psi_{17} - \psi_{22}\psi_{14} \\
a_{34} &= \psi_{11}\psi_{24} - \psi_{14}\psi_{21}
\end{aligned} \tag{45}$$

The coefficients in equation (36) are

$$\begin{aligned}
a_0 &= a_{10}a_{20} - a_{30}^2 \\
a_1 &= a_{11}a_{20} + a_{10}a_{21} - 2a_{31}a_{30} \\
a_2 &= a_{12}a_{20} + a_{11}a_{21} + a_{10}a_{22} - 2a_{32}a_{30} - a_{31}^2 \\
a_3 &= a_{13}a_{20} + a_{12}a_{21} + a_{11}a_{22} + a_{10}a_{23}
\end{aligned}$$

$$
\begin{aligned}
& -2a_{33}a_{30} - 2a_{31}a_{32} \\
a_4 &= a_{14}a_{20} + a_{13}a_{21} + a_{12}a_{22} + a_{11}a_{23} + a_{10}a_{24} \\
& -2a_{34}a_{30} - 2a_{31}a_{33} - a_{32}^2 \\
a_5 &= a_{14}a_{21} + a_{13}a_{22} + a_{12}a_{23} + a_{11}a_{24} \\
& -2a_{31}a_{34} - 2a_{32}a_{33} \\
a_6 &= a_{14}a_{22} + a_{13}a_{23} + a_{12}a_{24} - 2a_{34}a_{32} - a_{33}^2 \\
a_7 &= a_{14}a_{23} + a_{13}a_{24} - 2a_{34}a_{33} \\
a_8 &= a_{14}a_{24} - a_{34}^2
\end{aligned} \tag{46}
$$

9 Appendix 2

9.1 Coefficients of the rate equations

The terms obtained by differentiating the position loop closure equations are

$$
\begin{aligned}
K_{14} &= l_1^A s_{01}^A + l_2^A s_{012}^A + l_3^A s_{0123}^A \\
K_{15} &= l_2^A s_{012}^A + l_3^A s_{0123}^A \\
K_{16} &= l_3^A s_{0123}^A \\
K_{17} &= -l_1^B s_{01}^B - l_2^B s_{012}^B - l_3^B s_{0123}^B \\
K_{18} &= -l_2^B s_{012}^B - l_3^B s_{0123}^B \\
K_{19} &= -l_3^B s_{0123}^B \\
K_{24} &= -l_1^A c_{01}^A - l_2^A c_{012}^A - l_3^A c_{0123}^A \\
K_{25} &= -l_2^A c_{012}^A - l_3^A c_{012}^A \\
K_{26} &= -l_3^A c_{0123}^A \\
K_{27} &= l_1^B c_{01}^B + l_2^B c_{012}^B + l_3^B c_{0123}^B \\
K_{28} &= l_2^B c_{012}^B + l_3^B c_{0123}^B \\
K_{29} &= l_3^B c_{0123}^B
\end{aligned} \tag{47}
$$

The terms of the linerr momentum equations are

$$
\begin{aligned}
K_{41} &= m_0 + m_1^A + m_2^A + m_e + m_1^B + m_2^B = M \\
K_{43} &= My_0 \\
K_{44} &= -m_1^A l_{*1}^A s_{01}^A - m_2^A (l_1^A s_{01}^A + l_{*2}^A s_{012}^A) \\
K_{45} &= -m_2^A l_{*2}^A s_{012}^A \\
K_{47} &= -m_1^B l_{*1}^B s_{01}^B - m_2^B (l_1^B s_{01}^B + l_{*2}^B s_{012}^B) \\
& -m_e (l_1^B s_{01}^B + l_2^B s_{012}^B + l_3^B s_{0123}^B) \\
K_{48} &= -m_2^B l_{*2}^B s_{012}^B - m_e (l_2^B s_{012}^B + l_3^B s_{0123}^B) \\
K_{49} &= -m_e l_3^B s_{0123}^B \\
K_{52} &= M \\
K_{53} &= -Mx_0 \\
K_{54} &= m_1^A l_{*1}^A c_{01}^A + m_2^A (l_1^A c_{01}^A + l_{*2}^A c_{012}^A) \\
K_{55} &= m_2^A l_{*2}^A c_{012}^A \\
K_{56} &= 0 \\
K_{57} &= m_1^B l_{*1}^B c_{01}^B + m_2^B (l_1^B c_{01}^B + l_{*2}^B c_{012}^B)
\end{aligned}
$$

$$+m_e(l_1^B c_{01}^B + l_2^B c_{012}^B + l_3^B c_{0123}^B)$$
$$K_{58} = m_2^B l_{*2}^B c_{012}^B + m_e(l_2^B c_{012}^B + l_3^B c_{0123}^B)$$
$$K_{59} = m_e l_3^B c_{0123}^B \tag{48}$$

The terms of the angular momentum equation are

$$K_{63} = I_0 + m_1^A x_{*1}^A (l_0^A c_0 + l_{*1}^A c_{01}^A) +$$
$$m_1^A y_{*1}^A (l_0^A s_0 + l_{*1}^A s_{01}^A) + I_1^A +$$
$$m_2^A x_{*2}^A (l_0^A c_0 + l_1^A c_{01}^A + l_{*2}^A c_{012}^A) +$$
$$m_2^A y_{*2}^A (l_0^A s_0 + l_1^A s_{01}^A + l_{*2}^A s_{012}^A) + I_2^A$$
$$m_e x_e^B (l_0^B c_0 + l_1^B c_{01}^B + l_2^B c_{012}^B + l_3^B c_{0123}^B) +$$
$$m_e y_e^B (l_0^B s_0 + l_1^B s_{01}^B + l_2^B s_{012}^B + l_3^B s_{0123}^B) +$$
$$I_e + m_1^B x_{*1}^B (l_0^B c_0 + l_{*1}^B c_{01}^B) +$$
$$m_1^B y_{*1}^B (l_0^B s_0 + l_{*1}^B s_{01}^B) + I_1^B +$$
$$m_2^B x_{*2}^B (l_0^B c_0 + l_1^B c_{01}^B + l_{*2}^B c_{012}^B) +$$
$$m_2^B y_{*2}^B (l_0^B s_0 + l_1^B s_{01}^B + l_{*2}^B s_{012}^B) + I_2^B$$
$$K_{64} = m_1^A x_{*1}^A l_{*1}^A c_{01}^A + m_1^A y_{*1}^A l_{*1}^A s_{01}^A +$$
$$I_1^A + m_2^A x_{*2}^A (l_1^A c_{01}^A + l_{*2}^A c_{012}^A) +$$
$$m_2^A y_{*2}^A (l_1^A s_{01}^A + l_{*2}^A s_{012}^A) + I_2^A$$
$$K_{65} = m_2^A x_{*2}^A l_{*2}^A c_{012}^A + m_2^A y_{*2}^A l_{*2}^A s_{012}^A + I_2^A$$
$$K_{67} = m_1^B x_{*1}^B l_{*1}^B c_{01}^B + m_1^B y_{*1}^B l_{*1}^B s_{01}^B + I_1^B +$$
$$m_2^B x_{*2}^B (l_1^B c_{01}^B + l_{*2}^B c_{012}^B) + m_2^B y_{*2}^B (l_1^B s_{01}^B + l_{*2}^B s_{012}^B) +$$
$$I_2^B + m_e x_e^B (l_1^B c_{01}^B + l_2^B c_{012}^B + l_3^B c_{0123}^B) +$$
$$m_e y_e^B (l_1^B s_{01}^B + l_2^B s_{012}^B + l_3^B s_{0123}^B) + I_e$$
$$K_{68} = m_2^B x_{*2}^B l_{*2}^B c_{012}^b + m_2^B y_{*2}^B l_{*2}^B s_{012}^B +$$
$$I_2^B + m_e x_e^B (l_2^B c_{012}^B + l_3^B c_{0123}^B) +$$
$$m_e y_e^B (l_2^B s_{012}^B + l_3^B s_{0123}^B) + I_e$$
$$K_{69} = m_e x_e^B l_3^B c_{0123}^B + m_e y_e^B l_3^B s_{0123}^B + I_e \tag{49}$$

where

$$x_{*1}^A = x_0 + l_0^A c_0 + l_{*1}^A c_{01}^A$$
$$y_{*1}^A = y_0 + l_0^A s_0 + l_{*1}^A s_{01}^A$$
$$x_{*2}^A = x_0 + l_0^A c_0 + l_1^A c_{01}^A + l_{*2}^A c_{012}^A$$
$$y_{*2}^A = y_0 + l_0^A s_0 + l_1^A s_{01}^A + l_{*2}^A s_{012}^A$$
$$x_{*1}^B = x_0 + l_0^B c_0 + l_{*1}^B c_{01}^B$$
$$y_{*1}^B = y_0 + l_0^B s_0 + l_{*1}^B s_{01}^B$$
$$x_{*2}^B = x_0 + l_0^B c_0 + l_1^B c_{01}^B + l_{*2}^B c_{012}^B$$
$$y_{*2}^B = y_0 + l_0^B s_0 + l_1^B s_{01}^B + l_{*2}^B s_{012}^B$$
$$x_e^B = y_0 + l_0^B s_0 + l_1^B s_{01}^B + l_2^B s_{012}^B + l_3^B s_{0123}^B$$
$$y_e^B = y_0 + l_0^B s_0 + l_1^B s_{01}^B + l_2^B s_{012}^B + l_3^B s_{0123}^B \tag{50}$$

The terms in eqn (17) are

$$P_{13} = -l_0^A s_0 - l_1^A s_{01}^A - l_2^A s_{012}^A - l_3^A s_{0123}^A$$

$$
\begin{aligned}
P_{14} &= -l_1^A s_{01}^A - l_2^A s_{012}^A - l_3^A s_{0123}^A \\
P_{15} &= -l_2^A s_{012}^A - l_3^A s_{0123}^A \\
P_{16} &= -l_3^A s_{0123}^A \\
P_{23} &= l_0^A c_0 + l_1^A c_{01}^A + l_2^A c_{012}^A + l_3^A c_{0123}^A \\
P_{24} &= l_1^A c_{01}^A + l_2^A c_{012}^A + l_3^A c_{0123}^A \\
P_{25} &= l_2^A c_{012}^A + l_3^A c_{0123}^A \\
P_{26} &= l_3^A c_{0123}^A
\end{aligned} \tag{51}
$$

References

[1] Agrawal, S K and Ganmella, R ,"Workspace Boundaries of Free-floating Closed-Chain Planar Manipulators", In Proceedings, *Applied Mechanisms Conference*, Cincinnatti, 1991.

[2] Agrawal, S K and Rambhaskar, J ,"Jacobian Singularities of Free-floating Open-Chain Planar Manipulators", In Proceedings, *ASME Mechanism Conference*, Scottsdale, Arizona 1992

[3] Agrawal S.K and Shelly, M , 'Dynamic Simulation and Choice of Generalized Coordinates for a Free-floating Closed-Chain Planar Manipulators", to appear in *Mechanisms and Machine Theory*, 1992.

[4] Agrawal, S K. and Ganmella, R.,"A Path-planning Algorithm for a Free-floating Closed-Chain Planar Manipulators", In Proceedings, *ASME Mechanism Conference*, Scottsdale, Arizona, 1992.

[5] Alexander, H L and Cannon, R.H ,"An Extended Operational Space Control Algorithm for Satellite Manipulators", *The Journal of Astronautical Sciences*, vol 38, no 4, 1990, pp 473-486

[6] Ganmella, R., *Kinematics and Motion Planning of a Free-floating Closed-Chain Planar Manipulator*, M S Thesis, Department of Mechanical Engineering, Ohio University, 1992

[7] Longman,R W , Lindberg, R E , Zedd, M F , "Satellite Mounted Robot Manipulators - New Kinematics and Reaction Moment Compensation", *International Journal of Robotics Research*, vol 6, no 3, Fall 1987, pp 87-103

[8] Nakamura, Y and Mukherjee, R , "Nonholonomic Path Planning of Space Robots Via Bi-rectional Approach", *IEEE Transactions on Robotics and Automation*, vol 7, no 4, 1991, pp 500-514

[9] Papadopoulos, E and Dubowsky, S , "On the Dynamic Singularities in the Control of Space Manipulators", *ASME Winter Annual Meeting*, 1989, pp 45-51

[10] Papadopoulos, E and Dubowsky, S , "On the Nature of Control Algorithms for Space Manipulators", *IEEE International Conference on Robotics and Automation*, 1990, pp 1102-1108

[11] Schneider, S A and Cannon, R H ,"Object Impedance Control for Cooperative Manipulation Theory and Experimental Results", *IEEE Transactions on Robotics and Automation*, vol 8, no , June 1992, pp 383-394

[12] Sugimoto, K , Duffy, J and Hunt, K H , "Special Configurations of Spatial Mechanisms and Robot Arms,", *Mechanism and Machine Theory*, vol 17, no 2, 1982, pp 119-132

[13] Umetani, Y and Yoshida, K, "Continuous Path Control of Space Manipulators Mounted on OMV", *Acta Astronautica*, vol 15, no 12, 1987, pp 981-986

[14] Vafa, Z, "Space Manipulator with No Setellite Attitude Disturbance", *IEEE International Conference on Robotics and Automation*, 1990, pp. 1770-1775.

[15] Yoshida, K., Kurazume, R., and Umetani, Y, "Dual Arm Coordination in Space Free-flying Robot", In proceedings, *IEEE International Conference on Robotics and Automation*, vol 3, April 1991, pp 2516-2521.

OPTIMAL DESIGN AND LOCATION OF MANIPULATORS

Marco Ceccarelli

Dept. of Industrial Engineering
University of Cassino
Via Zamosch 43 - 03043 CASSINO (FR), Italy
tel. +39-776-299536 fax +39-776-310609

ABSTRACT - The problem of the manipulator design is approached, from a kinematic point of view, through an optimization formulation. The design variables are discussed including the location and the orientation of the robot base through suitable parameters which are congruent with the link parameters of the manipulator chain. A manipulator global length is defined as an objective function, since it may also take into account some workspace and inertial properties. Therefore, suitable procedures for the optimization problem are proposed. Some further considerations on optimal design procedures are developed by using an algebraic formulation for the manipulator workspace, which has been deduced in previous papers.

1. Introduction

The problem of the optimal design and location of manipulators have been addressed in the last decade, with particular formulations with respect to the optimization techniques and the robot structures.

In a pioneer paper, the minimization of manipulator proportions has been recognized and used to optimize also the layout of a manipulator workstation, (Scheinman and Roth, 1985).

However, the manipulator optimal design has been recognized and stated, basically, with respect to the workspace characteristics. This is because the workspace of a manipulator is one of its most important properties since it is related to the task area within which the robot manipulation can be performed.

A fundamental paper on the topics can be considered that one by Yang D.C.H. and Lee T.W., (Yang and Lee, 1984), in which the optimization problem has been formulated through a Volume Index on workspace and by using succesfully a heuristic technique, developed in the communications field.

Since the major complexity and computer time consumption is in the volume calculation, it has been thought convenient to adopt workspace formulations for specific manipulator structures and to develop a stepwise suitable procedure, (Tsai and Soni, 1985).

Recently, other investigators have approached the problem of the optimum workspace by adapting "ad hoc" formulations and procedures for the used optimization technique. Thus, the application of the Monte Carlo method has been explored to investigate its usefulness for the problems of kinematic synthesis of manipulators, (Rastegar and Fardanesh, 1990). A Sequential Quadratic Programming Technique has been succesfully applied to a specific optimization formulation for the three-revolute manipulator design, (Ceccarelli et al., 1992). A geometric reasoning has been also proposed to deduce optimization criteria for workspace volume and dexterity, which have been then applied to design a geometrically optimum six-revolute manipulator, (Vijaykumar et al., 1986). In addition, a specific optimization procedure for planar manipulators has been developed to obtain a workspace as close as possible to a prescribed one by using a

suitable description of the workspace boundary through a list of circular arcs and the application of the Gauss Divergence Theorem for the area calculation, (Gosselin and Guillot, 1991)

An algebraic approach has been proved to be an efficient and versatile tool in the numeric optimum workspace design for three-revolute manipulators, (Lin and Freudenstein, 1986) On this basis, an alternative algebraic formulation for the workspace boundary of three-revolute manipulators has been developed by the author with analysis purposes and to investigate the ring geometry, (Ceccarelli, 1989) The versatility of the algebraic approach has been further demonstrated by inverting the aforementioned formulation in a synthesis algorithm, (Ceccarelli, 1992). Moreover, this algebraic approach has been extended to four-revolute manipulators, (Ceccarelli and Vinciguerra, 1992), with the aim to achieve a general algebraic algorithm for workspace determination.

In this paper, an attempt is illustrated to formulate the design problem for manipulators by using this algebraic algorithm for the workspace determination. Particularly, an optimization formulation is proposed whose objective function is chosen as a global length of manipulators defined through the link parameters and the base location. This performance index has been proved to be significant on workspace and inertial properties of manipulators. Furthermore, the peculiarities of the proposed formulations for the optimization problem and the workspace determination give the possibility to design optimum manipulators by means of a search direction technique and through a suitable workspace precision points procedure.

2. Design Parameters of Manipulators

The optimal design of manipulator kinematic chains should include the optimal location of the robot base with respect to a fixed frame. Therefore, the design parameters can be considered the link time-independent sizes of the chain, i.e. for revolute-connected manipulators a_i, α_i and d_i, $i=1,...,n$, and the vectors $\underline{h}$ and $\underline{k}$, which represent respectively the position and the orientation of the robot base with respect to a fixed frame XYZ. Fig.1.

Since the main task of a manipulator is recognized in moving the end-effector and/or a grasped object in the space, it is suitable to achieve an optimal design and location of a robot taking into account basically workspace characteristics.

Moreover, also the encumbrance of the manipulator is recognized as an important issue since smaller and faster machinery and robots are desiderable in modern industrial applications.

Nevertheless, the two design features are not independent and a unique meaningful objective function in the optimization design problem can be formulated taking into account the following considerations.

It can be thought that the volume V of a manipulator workspace is related to the manipulator length L in the form

$$V = c\,L^{\beta} \tag{1}$$

where c is a function of the chain parameters; the manipulator length L can be defined as

$$L = \sqrt{\sum_1^n (a_i^2 + d_i^2)} \tag{2}$$

and β is a constant coefficient. In fact, it has been found that the workspace of revolute pairs connected manipulators depends on its main characteristics, as shape and volume, from the link ratios and dimensions, respectively, (Ceccarelli, 1989), (Ceccarelli and Vinciguerra, 1992). Therefore, it can be useful to express the manipulator length L in the form

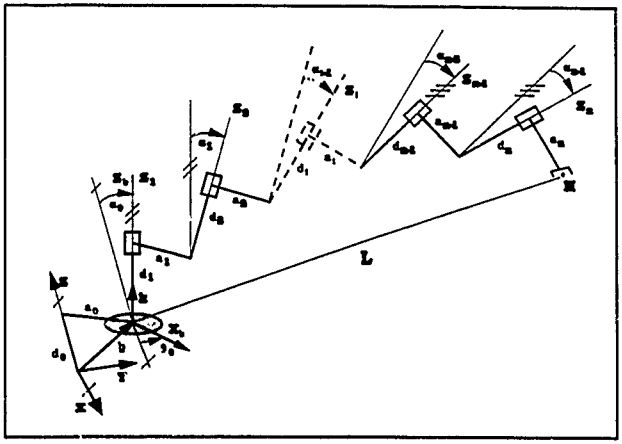


Fig.1. The design parameters for the kinematic chain of a manipulator.

$$L = a_1\sqrt{\sum_1^{2n+1} k_i^2} \tag{3}$$

where k_i , i=1....2n+1, are the link ratios of a_i and d_i , i=1....n, with respect to a reference dimension as for example a_1. Consequently, the workspace volume can be computed by introducing Eq (3) in the expression of the volume of a solid of revolution in the form

$$V = \frac{2\pi}{\sqrt{\sum_1^{2n+1} k_i^2}}\, L^3 \oint_{\delta W} r^{*2}\, dz^* \tag{4}$$

where r^* and z^* are the radial and axial reaches normalized with respect to a_1 and δW is the workspace boundary. In Eq (4) it is possible to recognize a shape factor and a ratio factor both influencing the workspace shape. L^3 represents a scaling factor giving the size of the workspace volume when the manipulator dimension is given through a_1 in Eq.(3). From the above expressions it can be observed that the size and the shape of a manipulator workspace are not completely independent in a sense that variations of link dimensions give arise to different changes of the manipulator length and the workspace volume.

Nevertheless, if the manipulator encumbrance increases the workspace volume will also increase. But sometimes the workspace volume may vanish even if L doesn't, since the workspace ring volume can

degenerate into a toroidal surface for particular values of the link parameters, as it has been stressed for the workspace ring geometry of three-revolute chains, (Ceccarelli, 1989). Furthermore, the minimum of the workspace volume can be the null value, but it is not related to the minimum encumbrance since it depends on the size of the degenerated toroidal surface. In this last case the two-revolute manipulator will be the optimal solution since its toroidal workspace can satisfy the same number of conditions for the workspace boundary surface as the case of three-revolute manipulators, (Ceccarelli, 1992), and probably it also occurs in the case of n-revolute manipulators

Consequently, it is thought convenient to assume as an objective function the "manipulator global lenght" $\mathcal{L}$ which can be defined in the form

$$\mathcal{L} = \sum_0^n (a_i + d_i) \tag{5}$$

where the robot base location has been included through the $\underline{h}$ vector expressed by means of the radial a_0 and axial d_0 components, Fig.1.

The angles α_0 and θ_0 can be considered as describing the orientation of the manipulator base as Z_b axis with respect to Z axis and X_b with respect to X, respectively. In addition, the base frame can be conveniently assumed to be parallel with the frame $X_1Y_1Z_1$, fixed on the first link of the manipulator chain, at a starting motion configuration. Therefore, the dimensional parameters for the optimal design problem will also include the parameters α_0, θ_0, a_0 and d_0.

The proposed formulation for the objective function can be thought meaningful also from a dynamical viewpoint since the term $(a_i + d_i)$ can be considered as a measure of inertial characteristics of the link, when the mass m_i and the inertial tensor I_i of the link i can be computed, respectively, as

$$m_i = \rho_i S_i (a_i + d_i) \tag{6}$$

$$I_i = \frac{\rho_i S_i}{6} \begin{vmatrix} 2d_i^3 & -3a_i d_i^2 s\alpha_i & -3a_i d_i^2 c\alpha_i \\ -3a_i d_i^2 s\alpha_i & 2a_i^3 + 6a_i^2 d_i + 2d_i^3 c^2\alpha_i & -2d_i^3 s\alpha_i c\alpha_i \\ -3a_i d_i^2 c\alpha_i & -2d_i^3 s\alpha_i c\alpha_i & 2a_i^3 + 6a_i^2 d_i + 2d_i^3 s^2\alpha_i \end{vmatrix}$$

where ρ_i is the mass density and S_i is the cross-section area.

Therefore, $\mathcal{L}$ represents a performance index describing synthetically both kinematic properties and inertial characteristics of a manipulator which can be considered as fundamental in a robot design, at least from a mechanical point of view.

3. An Optimization Formulation for the Design Problem of Manipulators

By considering $\mathcal{L}$ as an objective function, the design optimization problem for manipulators can be formulated in the form

$$\min \mathcal{L} \tag{7}$$

subject to

$$x_j \gtrless X_j \qquad j = 1,\dots J \tag{8}$$

$$V \geq V_0 \tag{9}$$

where χ_j (j=1.....J) represent given precision points, with respect to the fixed frame, to be reached as workspace points χ_j, V_0 is a minimum value for a desiderable workspace volume, and the design parameters are θ_0, a_i, α_i and d_i with i=0,1.....n.

The workspace precision points can be considered inside as well as on the boundary of the workspace volume. Nevertheless, they are assumed, at most, as limiting points for the workspace design capability and, conseguently, it is usually convenient to think them as workspace boundary points. In this case it is also possible to prescribe workspace characteristics as voids and hole, which can be useful in practical applications as save regions for equipment and personel in the automatized enviroment. Therefore, a workspace description through determination of its boundary is needed and χ_j is assumed to be determinable from an analytical expression of the workspace boundary. Once workspace boundary points are given, the manipulator solution must satisfy exactly the restriction with its workspace boundary and the sign equal is to be considered. Otherwise, a delimiting region can be assigned within which the workspace may be outlined and a weak restriction can be used to give larger design possibilities.

The constraint on the workspace volume can be of a determinant significance, since this restraint may have an effect on the robot size to counterbalance the minimization of L in order to ensure a no-degenerated solution of the manipulator chain.

The herein formulation, as well as the referenced optimal design procedures proposed in the last decade, is referred, properly speaking, to the dimensional synthesis of manipulator chains in a sense that the type of a manipulator structure cannot be a direct result of this optimal design, but it is better considered as a datum. However, this usually can be obtained by considering the number of degrees of freedom which is required for a specific manipulation task. Nevertheless, a certain design optimal selection can be achieved by comparing different manipulator structures through the results of the same optimization problem.

4. A Workspace Formulation

In order to facilitate the numerical solution of the design problem (7), it can be useful to express the envolved workspace characteristics by means of a proper analytical formulation.

In the paper, the case of three-revolute manipulators is illustrated in detail, since most of the robot arms show this kinematic structure to produce the gross motion.

The ring workspace geometry of a three-revolute manipulator can be described by considering a generation process of the workspace boundary through a suitable torus family envelope. Thus, an algebraic formulation can be deduced and expressed through the radial reach r_b and the axial reach z_b of the workspace boundary points with respect to the base frame in the form, (Ceccarelli, 1989),

$$\begin{aligned} r_b &= r_1 \\ z_b &= z_1 + d_1 \end{aligned} \tag{10}$$

in which the radial distances r_1 and z_1 with respect to Z_1 are given, with the hypotheses $C_1 \neq 0$ and $E_1 \neq 0$ as

$$r_1 = \left[A_1 - z_1^2 + \frac{(C_1 z_1 + D_1) G_1 + F_1}{E_1} \right]^{1/2} \tag{11}$$

$$z_1 = \frac{-L_1 \pm Q_1^{1/2}}{K_1 C_1} - \frac{D_1}{C_1}$$

where the structural coefficients can be computed as

$$\begin{aligned}
A_1 &= a_1^2 + r_2^2 + (z_2 + d_2)^2 \\
B_1 &= -4\, a_1^2\, r_2^2 \\
C_1 &= 2\, a_1 / s\alpha_1 \\
D_1 &= -2\, a_1 (z_2 + d_2)\, c\alpha_1 / s\alpha_1 \\
E_1 &= -2\, a_3 (d_2\, s\alpha_2\, c\theta_3 + a_2\, s\theta_3) \\
F_1 &= 4\, a_1^2\, a_3 (a_3\, s\alpha_2^2\, s\theta_3\, c\theta_3 + a_2\, s\theta_3 - d_3\, s\alpha_2\, c\alpha_2 c\theta_3) \\
G_1 &= 2\, a_1\, a_3\, c\alpha_1\, s\alpha_2\, c\theta_3 / s\alpha_1 \\
K_1 &= G_1^2 + E_1^2 \\
L_1 &= F_1\, G_1 \\
Q_1 &= L_1^2 - K_1 (F_1^2 + E_1 E_1^2)
\end{aligned} \tag{12}$$

and the reach distances r_2 and z_2 can be expressed as

$$\begin{aligned}
r_2 &= [(a_3\, c\theta_3 + a_2)^2 + (a_3\, s\theta_3\, c\alpha_2 + d_3\, s\alpha_2)^2]^{1/2} \\
z_2 &= d_3\, c\alpha_2 - a_3\, s\theta_3\, s\alpha_2
\end{aligned} \tag{13}$$

The angle θ_3 is the joint angle of the extreme pair in the chain and it is the kinematic time-dependent variable for the workspace determination. In fact, the workspace boundary can be obtained from Eqs.(11)-(13) by scanning θ_3 from 0 to 2π, (Ceccarelli, 1989).

This workspace formulation can be manipulated with synthesis purposes to give the workspace boundary points through the position vector $\underline{x}$ with respect to a fixed frame, (Ceccarelli, 1992).

$$E_1(\underline{x}\cdot\underline{x} - 2\underline{b}\cdot\underline{x} + \underline{b}\cdot\underline{b} - A_1) - \{C_1[\underline{r}_3\cdot(\underline{x}-\underline{b})] + D_1\}G_1 + F_1 = 0 \tag{14}$$

$$\underline{r}_3\cdot(\underline{x}-\underline{b}) = \frac{-L_1 \pm Q_1^{1/2}}{K_1 C_1} - \frac{D_1}{C_1}$$

where $\cdot$ is the symbol for the dot product, $\underline{x}$ is the vector of a workspace boundary point computed from the vector $\underline{x}_b$, whose components are r_b and z_b, by means of the expression of reference change $\underline{x}_b = R\,\underline{x} - \underline{b}_b$ by using the rotation matrix R and the base position vector $\underline{b} = R^t\,\underline{b}_b$ (t represents the transposition operator); $\underline{r}_3$ is the third row vector in R which gives the line direction of Z with respect to $X_b Y_b Z_b$.

Eqs.(10)-(14) are useful to express analytically the constraint Eqs.(8) and (9) as explicit functions of the design parameters. In addition, these equations are useful to calculate analytically the derivatives which are involved in the numerical optimization process. These derivatives can be computed in a closed-

form as a function of the design parameters from the above mentioned expressions. For example, in the case of the a_1 design parameter, the first derivatives can be taken the form

$$\frac{\partial r_{1j}}{\partial a_1}=\frac{1}{2r_{1j}}\left[\frac{\partial A_1}{\partial a_1}-2z_{1j}\frac{\partial z_{1j}}{\partial a_1}-\frac{(C_1z_{1j}+D_1)G_1+F_1}{E_1^2}\frac{\partial E_1}{\partial a_1}\right] + \frac{1}{2r_{1j}E_1}\left[\frac{\partial F_1}{\partial a_1}+\frac{\partial G_1}{\partial a_1}(C_1z_{1j}+D_1)+G_1\left(\frac{\partial D_1}{\partial a_1}+z_{1j}\frac{\partial C_1}{\partial a_1}+C_1\frac{\partial z_{1j}}{\partial a_1}\right)\right] \tag{15}$$

$$\frac{\partial z_{1j}}{\partial a_1}=-\frac{z_{1j}}{C_1}\frac{\partial C_1}{\partial a_1}-\frac{-L_1\pm Q_1^{1/2}}{K_1^2C_1}\frac{\partial K_1}{\partial a_1}+\frac{1}{K_1C_1}\left(\pm\frac{1}{2Q_1}\frac{\partial Q_1}{\partial a_1}-\frac{\partial L_1}{\partial a_1}-K_1\frac{\partial D_1}{\partial a_1}\right) \tag{16}$$

and

$$\frac{\partial u_j}{\partial a_1}=\mp\frac{1}{2}\left[\frac{\partial A_1}{\partial a_1}-\left(\frac{-L_1\pm Q_1^{1/2}}{E_1^2K_1}+\frac{F_1}{E_1^2}\right)\frac{\partial E_1}{\partial a_1}-\frac{-L_1\pm Q_1^{1/2}}{K_1^2E_1}\frac{\partial K_1}{\partial a_1}+\frac{1}{E_1}\frac{\partial F_1}{\partial a_1}\right] -\frac{1}{2K_1E_1}\left(\frac{\partial L_1}{\partial a_1}\pm\frac{1}{2Q_1^{1/2}}\frac{\partial Q_1}{\partial a_1}\right) \tag{17}$$

where $r_{1,j}$ and $z_{1,j}$ are the reach coordinates of $\underline{x}_{1,j}$ with respect to the link 1 frame in the constraints equations (8) and (9); u_j, v_j and w_j are cartesian components of $\underline{x}_{1,j}$; and the derivatives of the structural coefficients can be evaluated by means of the definition equations (12) and (13). Higher derivatives can be obtained by further differentiating the above expressions and using the workspace algebraic formulation.

A further useful development of the algebraic approach concerns with the volume calculation. In fact, the ring geometry of a workspace of revolute pair connected manipulators allows to formulate an efficient numeric expression for volume evaluation by considering a limited number of workspace boundary points through Eqs.(11)-(13). Thus, the volume V can be calculated by summing up, algebraically, the volumes of thin cylinders individuated, each one, by two boundary points n and n+1 in the form, (Ceccarelli and Vinciguerra, 1992),

$$V=\sum_{n}^{N}\frac{\pi}{2}\left(z_{1,n+1}+z_{1,n}\right)\left(r_{1,n}^2-r_{1,n+1}^2\right) \tag{18}$$

in which $r_{1,n}$ and $z_{1,n}$ are computed from Eqs.(11)-(13) through an N point discretization in θ_3, when the manipulator proportions are given. The volume of voids will be given with a negative sign so that it is subtracted within Eq (18), when a suitable choice of positive sign in a listing of the N number points is adopted. Moreover, the derivative of the workspace volume with respect to the design parameters can be

computed by differentiating Eq (18). For example, the derivative with respect to a_1 can be obtained in the form

$$\frac{\partial V}{\partial a_1} = \sum_i^N \frac{\pi}{2}\left[\left(\frac{\partial z_{i,n+1}}{\partial a_1}+\frac{\partial z_{i,n}}{\partial a_1}\right)\left(r_{i,n}^2 - r_{i,n+1}^2\right)+2\left(z_{i,n+1}+z_{i,n}\right)\left(r_{i,n}\frac{\partial r_{i,n}}{\partial a_1}-r_{i,n+1}\frac{\partial r_{i,n+1}}{\partial a_1}\right)\right] \quad (19)$$

in which the radial and axial reach derivatives with respect to a_1 can be calculated for the actual boundary discretization in θ_3, by using once more Eqs. (15) and (16).

5. Procedures for an Optimal Design

Several optimization techniques can be available to solve the design optimization problem formulated by Eqs.(7)-(9). Nevertheless, the peculiarities of the formulation and the proposed algebraic algorithm for the determination of the workspace characteristics may give arise to particular considerations and developments.

When the workspace points X_j , j=1,....J, are given as precision workspace boundary points, then the optimal design can be better achieved by approaching the problem through a suitable set of algebraic equations deduced from Eqs. (11) or (14). This procedure can be developed when the precision point requirements are thought to be crucial and of prior importance or so severe that the optimization problem may not have solution. In this case, Eqs.(8) can be written as equality equations and a suitable procedure using the workspace formulation of Eqs.(14) can be developed to solve the design and location problem of manipulators. In particular, once Eqs.(14) are solved numerically to give the structural coefficients and the base location vectors, then the dimensional parameters of the chain can be obtained algebraically from the definition Eqs.(12), (Ceccarelli, 1992). Hence, an optimal solution, among all the obtainable feasible designs, can be achieved by choosing that one which has the minimum global length. In this case, indeed, the constraint (9) on the workspace volume may not be satisfied, and particular care must be taken on its significance and requirement, since the workspace boundary points already impose, practically, a certain workspace volume.

When the design requirements are given to obtain a workspace within a prescribed working area through limitations on the reach possibilities, then a proper optimization problem with Eqs.(7)-(9) can be solved. Eqs.(10)-(19) can be again used with great computational efficiency in a suitable procedure, within a numerical search direction optimization technique, for the optimal design of manipulators. In Fig.2 a flowchart for the optimal design of manipulators is illustrated by stressing the importance and the usefulness of the proposed algebraic formulation for the workspace evaluation. In addition, the algorithm has been developed so that a control on the iterative solution is obtained through suitable convergence parameters for the obtained values of the objective function, the workspace precision points and the workspace volume. By assuming greater or smaller values of these convergence parameters, an optimal design can be adjusted to a practical design in a sense of speeding up the numerical process by relaxing or forcing, respectively, the design requirements.

An example is illustrated in Fig.3. The optimization is started with an initial manipulator whose workspace is quite far to satisfy the design requirements which are indicated with black circles. In particular, the small black circles indicate points to be crossed or overpassed by the workspace boundary and the greater the limit of a prescribed working area on radial reach. The ground has been assumed on zero axial coordinate so that the workspace is required to be outlined not below $z_b = 0$, with a workspace volume near as possible to $1000\ a_1^3$. In this case, the robot base has been considered given to stress the efficiency of the algebraic formulation. The results of the different iterations are shown by a fine line, and the final result by a heavier line. It can be noted that although the starting manipulator workspace does not satisfy anyone of the design requirements, nevertheless the algorithm reaches a feasible solution with

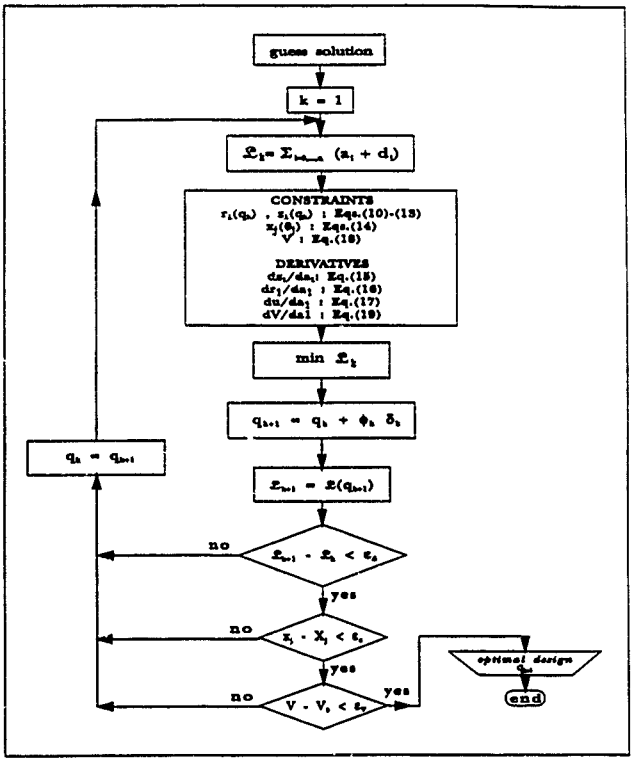


Fig.2. A design procedure scheme, through an algebraic formulation of the workspace boundary, for the optimization problem applied to robotic manipulators, in which q_k is the design parameter vector at step k; ε_L, ε_Σ and ε_V are the numerical precision factors with respect to the design indices L, Σ_j and V; δ_k and ϕ_k are the search direction vector and the scalar step in an optimization numerical process

few iterations and it optimizes the workspace with a manipulator length of 11.89 a_1 and a volume of 1000.34 a_1^3

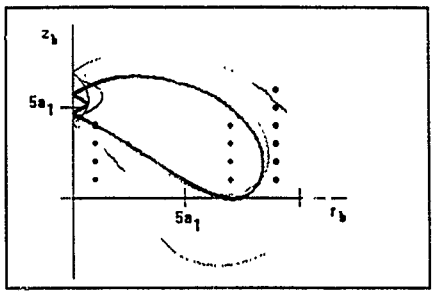


Fig.3. An example of optimal design evolution of three-revolute manipulator workspace.

The proposed algorithm can be extended to n-revolute manipulator chains with no great efforts by using an extension of the formulation of Eqs (8)-(10). As an explanatory example can be illustrated the case of four-revolute manipulators by using the workspace evaluation formulation proposed in (Ceccarelli and Vinciguerra, 1992). In this case, in fact, a generalization of the workspace formulation can be deduced by considering that each boundary point of a three-revolute manipulator workspace will trace a torus when the first two joints in the chains are rotated. Therefore, a torus family can be obtained from all the points of the workspace boundary so that a suitable envelope algorithm can determinate the workspace boundary of four-revolute manipulators, (Ceccarelli and Vinciguerra, 1992), through yet the expression of Eqs (11), in which further structural coefficients are considered in the form

$$\begin{aligned} E_1 &= R_2 + S_2 \\ F_1 &= -2a_1^2 R_2 \\ G_1 &= -2a_1 z_2' c\alpha_1 / s\alpha_1 \\ L_1 &= F_1 G_1 \\ K_1 &= E_1^2 + G_1^2 \\ Q_1 &= L_1^2 - K_1 (F_1^2 + B_1 E_1^2) \end{aligned} \tag{20}$$

and

$$R_2 = \frac{(C_2 z_2 + D_2)(E_2 G_2 - G_2 E_2') + E_2 F_2 - F_2 E_2' + G_2 E_2 (C_2 z_2' + G_2)}{E_2^2} + E_2 - 2 z_2 z_2'$$

$$S_2 = 2(z_2 + d_2) z_2' \tag{21}$$

$$z_2' = \frac{(\pm 0.5 Q_2' Q_2^{-1/2} - L_2')K_2 - K_2'(-L_2 \pm Q_2^{-1/2})}{C_2 K_2^2} \frac{G_2}{C_2}$$

where r_2, z_2 and the involved coefficients are now expressed by means of Eqs (11)-(13) by an increase of one for all the pedices In this way, it is still possible to calculate the workspace volume and its derivative through Eqs (18) and (19), the formulation of Eqs.(14) is yet describing the workspace boundary points with respect to a fixed frame, and the derivatives of the position vectors of workspace points can be expresse as the Eqs (15)-(17). Therefore, the optimal design procedure of the flowchart in Fig.2 can be applied with small algebraic adjustments only in the workspace algorithm.

6. Conclusion

In this paper, the design problem of manipulators is formulated as an optimization problem by using a proper algebraic formulation for the workspace boundary. This algebraic approach has been further developed to facilitate the numerical solution of an optimal design by proposing suitable procedures and by deducing a straightforward algorithm for the calculation of the workspace volume

Furthermore, the position and the orientation of the robot base with respect to a fixed frame have been expressed in the form of link parameters so that a manipulator global length, defined as a sum of all these dimensional parameters, has been assumed as an objective function. Thus, a general formulation does not differ from the particular case with assigned location of the robot base. In addition, the proposed performance index is meaningful for optimization also from a mechanical point of view, since it has been shown to be significant for both workspace and inertial properties of manipulator chains

Some aspects of the numerical solving process are discussed referring to the three-revolute manipulators. Nevertheless, the algorithms can be extended, in future developments, to n-revolute manipulator chains by using an extension of the workspace formulation, which is underway at the Department of Industrial Engineering at University of Cassino. However, the basic concept has been synthetically illustrated in this paper by examining the formulation for the optimal design of a four-revolute manipulator.

7. References

Ceccarelli M. (1989) On the Workspace of 3R Robot Arms. *Proceedings of the Fifth IFToMM International Symposium on Theory and Practice of Mechanisms*, Bucharest, 1989, Vol.II-1, 37-46

Ceccarelli M (1992). A Synthesis Algorithm of Three-Revolute Manipulators by means of the Workspace Contour Algebraic Formulation *Proceedings of the ASME 22nd Biennal Mechanisms Conference*, Scottsdale, DE-Vol.45, 397-403.

Ceccarelli M., and A. Vinciguerra (1992) Workspace Evaluation of General 4R Manipulators *Proceedings of 3rd International Workshop on Advances in Robot Kinematics*, Ferrara, 313-317

Ceccarelli M., Mata V., and F. Valero (1992). An Optimization Formulation of Three-Revolute Manipulator Synthesis. *Proceedings of the XI AIMETA National Congress*, Trento, Vol Theory of Machines, 91-96

Lin C.D., and F Freudenstein (1986). Optimization of the Workspace of a Three-Link Turning-Pair Connected Robot Arm. *The Int Jnl of Robotics Research*, n 2, 104-111

Rastegar J., and B Fardanesh (1990). Geometric Synthesis of Manipulators Using the Monte Carlo Method *ASME Jnl of Mechanical Design*, 112, 450-452.

Scheinman V., and b. Roth (1985) On the Optimal Selection and Placement of Manipulators *Proceedings of the Fifth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, Kogan Page, London, 39-45.

Tsai Y C., and A H. Soni (1985). Workspace Synthesis of 3R, 4R, 5R and 6R Robots *IFToMM Jnl Mechanism and Machine Theory*, n 6, 555-563

Yang, D C.H., and T.W Lee (1984) Heuristic Combinatorial Optimization in the Design of Manipulator Workspace *IEEE Transactions on Systems Man and Cybernetics*, SMC-14, n 4, 571-580.

8. Acknowledgements

This research received a financial support from the NATO ASI, that is gratefully acknowledged

THEORETICAL MODELING, NONLINEAR CONTROL AND SENSITIVITY ANALYSIS OF A SPATIAL MULTI-AXIS TEST FACILITY BASED ON COMPUTER ALGEBRA

K.-D. Leimbach, H. Hahn
Department of Mechanical Engineering (FB 15), University of Kassel
3500 Kassel, Mönchebergstraße 7,
Germany

ABSTRACT. This paper presents a nonlinear control concept of a spatial multi-axis servo-hydraulic test facility together with a parameter sensitivity analysis of the closed loop system. Based on nonlinear model equations including the servohydraulic actuator dynamics and the test table and payload mechanics a global nonlinear diffeomorphism is derived which maps the model equations into nonlinear canonical form. Using symbolic languages, a nonlinear control law is developed and calculated based on exact linearization techniques. The efficiency of this control concept with respect to tracking and decoupling behaviour of the test facility is demonstrated by computer simulations, taking into account variations of mechanical and hydraulic plant parameters. The applicability of this control approach to test facilities from industrial practice is discussed in detail.

1. Introduction

High quality multi-axis servohydraulic test facilities are widely used for testing of critical components of industrial equipment and of future spacecraft (Hahn, 1986). Theoretical investigations of multi axis test facility control concepts are usually based on simplified linear model equations (Hahn, 1986, 1992). However, the exact model equations of such systems are highly nonlinear and strongly coupled. This paper provides a nonlinear control concept for a spatial multi-axis servohydraulic test facility (cf. 1) based on exact input state linearization of the control plant equations (Isidori, 1989, Slotine, 1991; Schwarz, 1991). Based on the nonlinear control plant equations of Section 2 including the mechanical equations of motion of the test table and payload and the servohydraulic actuator models, a nonlinear control concept is developed in Section 3. The extreme lengthy analytical expressions occurring in the design steps

- test of exact linearization conditions,
- construction of a suitable diffeomorphism,
- mapping on the control plant equations of motion to nonlinear normal form and
- calculation of the nonlinear controllers

have been reliably derived and handled by systematically using symbolic languages. The sensitivity of the test facility behaviour with respect to critical (mechanical and hydraulic) plant parameters is discussed in Section 4. The practical use of this approach for a coordinated control of spatial motions of test facilities is discussed in Section 5.

2. Nonlinear Model Equations of the Plant Mechanics and Hydraulics

The test facility considered includes the following components (cf. 1)

- a rigid six degree of freedom test table with a rigidly attached rigid payload
- six servohydraulic actuators and
- an integrated control system

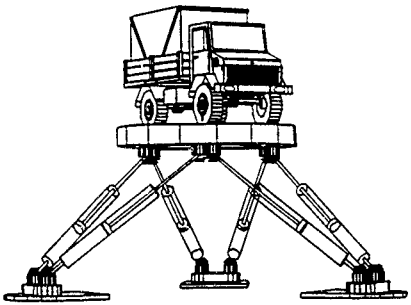


Figure 1: Computer drawing of a multi-axis test facility driven by six actuators

2.1 TEST TABLE AND PAYLOAD MECHANICS

The nonlinear model equations of the test facility mechanics are

$$M(x_1)\cdot \dot{x}_2 + J_{z1}^T(x_1)\; D_k\; J_{z1}(x_1)\; x_2 + n(x_1,x_2) + q_G(x_1) = J_{z1}^T(x_1)\; A_k\; p \qquad (1)$$

(cf. 2) where $M(x)$ is the inertia matrix of the test facility

$$M(x_1) = \begin{bmatrix} \begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{pmatrix} & m \begin{pmatrix} 0 & -z_{CP}^L & y_{CP}^L \\ z_{CP}^L & 0 & -x_{CP}^L \\ -y_{CP}^L & x_{CP}^L & 0 \end{pmatrix} \begin{pmatrix} c_2c_3 & c_1s_3+s_1s_2c_3 & s_1s_3-c_1s_2c_3 \\ -c_2s_3 & c_1c_3-s_1s_2s_3 & s_1c_3+c_1s_2s_3 \\ s_2 & -s_1c_2 & c_1c_2 \end{pmatrix} \cdot \underbrace{\begin{pmatrix} J_{xx}^L & -J_{xy}^L & -J_{xz}^L \\ -J_{xy}^L & J_{yy}^L & -J_{yz}^L \\ -J_{xz}^L & -J_{yz}^L & J_{zz}^L \end{pmatrix}}_{=J_C^L} \\ \cdot \begin{pmatrix} c_2c_3 & -c_2s_3 & s_2 \\ c_1s_3+s_1s_2c_3 & c_1c_3-s_1s_2s_3 & -s_1c_2 \\ s_1s_3-c_1s_2c_3 & s_1c_3+c_1s_2s_3 & c_1c_2 \end{pmatrix} \begin{pmatrix} 0 & z_{CP}^L & -y_{CP}^L \\ -z_{CP}^L & 0 & x_{CP}^L \\ y_{CP}^L & -x_{CP}^L & 0 \end{pmatrix} & \\ m \begin{pmatrix} 0 & -z_{CP}^L & y_{CP}^L \\ z_{CP}^L & 0 & -x_{CP}^L \\ -y_{CP}^L & x_{CP}^L & 0 \end{pmatrix} \begin{pmatrix} 0 & z_{CP}^L & -y_{CP}^L \\ -z_{CP}^L & 0 & x_{CP}^L \\ y_{CP}^L & -x_{CP}^L & 0 \end{pmatrix} & m \begin{pmatrix} \left(y_{CP}^{L\,2}+z_{CP}^{L\,2}\right) & -x_{CP}^L y_{CP}^L & -x_{CP}^L z_{CP}^L \\ -y_{CP}^L x_{CP}^L & \left(x_{CP}^{L\,2}+z_{CP}^{L\,2}\right) & -y_{CP}^L z_{CP}^L \\ -z_{CP}^L x_{CP}^L & -z_{CP}^L y_{CP}^L & \left(x_{CP}^{L\,2}+y_{CP}^{L\,2}\right) \end{pmatrix} \end{bmatrix} \qquad (2)$$

$r_{CP}^L = \left[x_{CP}^L,\, y_{CP}^L,\, z_{CP}^L\right]^T$ is the vector from a reference point P to the center of gravity C represented in local frame L, m is the test facility mass; J_C^L is the inertia tensor of the test facility,

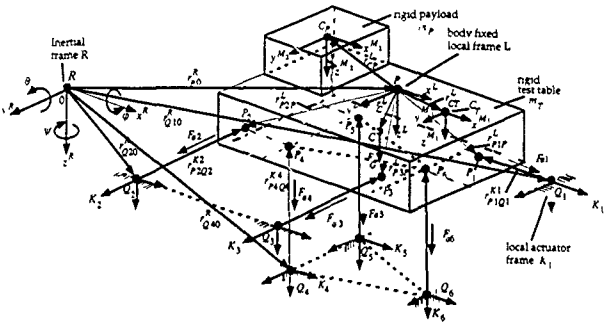


Figure 2 Multi-axis test facility with six servohydraulic actuators (definitions of vectors and frames)

φ, θ, ψ are the Bryant angles ($s_1 = \sin(\varphi)$, $s_2 = \sin(\theta)$, $s_3 = \sin(\psi)$, $c_1 = \cos(\varphi)$, $c_2 = \cos(\theta)$, $c_3 = \cos(\psi)$);

$$D_K = diag(d_{K1}, \ldots, d_{K6}) \quad \text{and} \quad A_K = diag(A_{K1}, \ldots, A_{K6}) \tag{3}$$

are the diagonal matrices of the actuator damping coefficients d_{Ki} and of the piston areas A_{Ki}, respectively, x_1 is the position and orientation vector of the spatial test facility represented in inertial frame R; x_2 and $\dot{x}_2$ are the velocity and acceleration vectors, respectively,

$$x_1 = \left[x_P^R, y_P^R, z_P^R, \varphi, \theta, \psi \right]^T, \quad x_2 = \left[\dot{x}_P^R, \dot{y}_P^R, \dot{z}_P^R, p^L, q^L, r^L \right]^T \in R^6, \tag{4}$$

where the relation among variables p^L, q^L, r^L and φ, θ, ψ is defined in (14) and (15) The nonlinear vector $n(x_1, x_2)$ of the centrifugal forces and of the gyroscopic terms is defined in (5),

$$n(x_1, x_2) = \begin{bmatrix} m \begin{pmatrix} c_2c_3 & -c_2s_3 & s_2 \\ c_1s_3 + s_1s_2c_3 & c_1c_3 - s_1s_2s_3 & -s_1c_2 \\ s_1s_3 - c_1s_2c_3 & s_1c_3 + c_1s_2s_3 & c_1c_2 \end{pmatrix} \begin{pmatrix} -\left(r^{L2} + q^{L2}\right) x_{CP}^L + p^L q^L y_{CP}^L + p^L r^L z_{CP}^L \\ p^L q^L x_{CP}^L - \left(p^{L2} + r^{L2}\right) y_{CP}^L + q^L r^L z_{CP}^L \\ p^L r^L x_{CP}^L + q^L r^L y_{CP}^L - \left(p^{L2} + q^{L2}\right) z_{CP}^L \end{pmatrix} \\ \begin{pmatrix} 0 & -r^L & q^L \\ r^L & 0 & -p^L \\ -q^L & p^L & 0 \end{pmatrix} \left[\begin{pmatrix} J_{Cx}^L & -J_{Cxy}^L & -J_{Cxz}^L \\ -J_{Cyx}^L & J_{Cy}^L & -J_{Cyz}^L \\ -J_{Czx}^L & -J_{Czy}^L & J_{Cz}^L \end{pmatrix} + m \begin{pmatrix} \left(y_{CP}^{L2} + z_{CP}^{L2}\right) & -x_{CP}^L y_{CP}^L & -x_{CP}^L z_{CP}^L \\ -y_{CP}^L x_{CP}^L & \left(x_{CP}^{L2} + z_{CP}^{L2}\right) & -y_{CP}^L z_{CP}^L \\ -z_{CP}^L x_{CP}^L & -z_{CP}^L y_{CP}^L & \left(x_{CP}^{L2} + y_{CP}^{L2}\right) \end{pmatrix} \right] \begin{pmatrix} p^L \\ q^L \\ r^L \end{pmatrix} \end{bmatrix} \tag{5}$$

$q_G(x)$ is the nonlinear vector of gravitational force and torque's

$$q_G(x_1) = \left[\begin{pmatrix} 0 & -z^L_{CP} & y^L_{CP} \\ z^L_{CP} & 0 & -x^L_{CP} \\ -y^L_{CP} & x^L_{CP} & 0 \end{pmatrix} \begin{pmatrix} c_2c_3 & c_1s_3+s_1s_2c_3 & s_1s_3-c_1s_2c_3 \\ -c_2s_3 & c_1c_3-s_1s_2s_3 & s_1c_3+c_1s_2s_3 \\ s_2 & -s_1c_2 & c_1c_2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -m\,g \end{pmatrix} \right] \quad (0\,,\,0\,,\,-m\,g)^T \quad (6)$$

and $$p = [p_1\,,\,p_2\,,\,p_3\,,\,p_4\,,\,p_5\,,\,p_6]^T \in R^6 \quad (7)$$

is the vector of the actuator pressure differences. The nonlinear transformation matrices $J^T_{x1}(x_1)$ and $J_{x1}(x_1)$ map the test table forces and velocities from joint fixed frames K_1 to test table degrees of freedom representation in frame L and vice versa, where

$$J_{x1}(x_1) = \begin{bmatrix} \mathrm{Pr}x\,A^{K1R} & -\mathrm{Pr}x\,A^{K1R}\,A^{RL}\,\tilde{r}^L_{P1P} \\ \mathrm{Pr}y\,A^{K2R} & -\mathrm{Pr}y\,A^{K2R}\,A^{RL}\,\tilde{r}^L_{P2P} \\ \mathrm{Pr}y\,A^{K3R} & -\mathrm{Pr}y\,A^{K3R}\,A^{RL}\,\tilde{r}^L_{P3P} \\ \mathrm{Pr}z\,A^{K4R} & -\mathrm{Pr}z\,A^{K4R}\,A^{RL}\,\tilde{r}^L_{P4P} \\ \mathrm{Pr}z\,A^{K5R} & -\mathrm{Pr}z\,A^{K5R}\,A^{RL}\,\tilde{r}^L_{P5P} \\ \mathrm{Pr}z\,A^{K6R} & -\mathrm{Pr}z\,A^{K6R}\,A^{RL}\,\tilde{r}^L_{P6P} \end{bmatrix} \in R^{6\,6} \quad \text{and} \quad (8)$$

$$A^{RL} = \begin{bmatrix} c_2c_3 & -c_2s_3 & s_2 \\ c_1s_3+s_1s_2c_3 & c_1c_3-s_1s_2s_3 & -s_1c_2 \\ s_1s_3-c_1s_2c_3 & s_1c_3+c_1s_2s_3 & c_1c_2 \end{bmatrix} \quad (9)$$

is the orientation matrix of the test facility fixed frame L with respect to the inertial frame R. The projection vectors are defined as $\mathrm{Pr}x = [1\,,0\,,0]$, $\mathrm{Pr}y = [0\,,1\,,0]$, , $\mathrm{Pr}z = [0\,,0\,,1]$, r^L_{PiP} $(i = 1,\ \ ,6)$ is the vector from reference point P to the test table fixed actuator joint P_i, where

$$r^L_{PiP} = [x^L_{PiP},\, y^L_{PiP},\, z^L_{PiP}]^T, \quad \tilde{r}^L_{PiP} = \begin{bmatrix} 0 & -z^L_{PiP} & y^L_{PiP} \\ z^L_{PiP} & 0 & -x^L_{PiP} \\ -y^L_{PiP} & x^L_{PiP} & 0 \end{bmatrix}, \ (i = 1,\ \ ,6) \quad (10)$$

The actuator orientation matrices A^{KiR} are described exemplary by matrix A^{K1R} associated to the x-actuator (cf. 3 and (11)).

$$A^{K1R} = \begin{bmatrix} \cos\gamma_1\cos\beta_1 & \sin\gamma_1 & -\cos\gamma_1\sin\beta_1 \\ -\sin\gamma_1\cos\beta_1 & \cos\gamma_1 & \sin\gamma_1\sin\beta_1 \\ \sin\beta_1 & 0 & \cos\beta_1 \end{bmatrix} \quad (11)$$

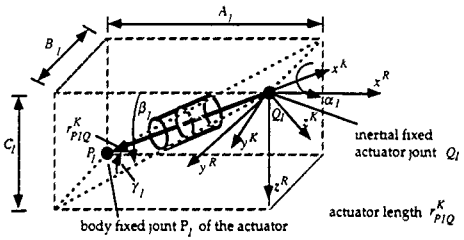


Figure 3 Orientation of the joint fixed frame K_1 of the x-actuator

where (cf 3)

$$\sin\beta_1 = \frac{C_1}{\sqrt{A_1^2 + C_1^2}} \quad , \quad \cos\beta_1 = \frac{-A_1}{\sqrt{A_1^2 + C_1^2}} \quad ,$$
$$\sin\gamma_1 = \frac{-B_1}{\sqrt{A_1^2 + B_1^2 + C_1^2}} \quad , \quad \cos\gamma_1 = \frac{\sqrt{A_1^2 + C_1^2}}{\sqrt{A_1^2 + B_1^2 + C_1^2}} \quad , \tag{12}$$

$$\begin{aligned} A_1 &= x_P^R + c_2\ c_3\ x_{P1P}^L - c_2 \cdot s_3\ y_{P1P}^L + s_2\ z_{P1P}^L - x_{Q10}^R \quad , \\ B_1 &= y_P^R + (c_1\ s_3 + s_1\ s_2\ c_3)\ x_{P1P}^L + (c_1\ c_3 - s_1 \cdot s_2\ s_3)\ y_{P1P}^L - s_1\ c_2\ z_{P1P}^L - y_{Q10}^R \quad , \\ C_1 &= z_P^R + (s_1\ s_3 - c_1\ s_2\ c_3)\ x_{P1P}^L + (s_1 \cdot c_3 + c_1\ s_2\ s_3)\ y_{P1P}^L + c_1\ c_2\ z_{P1P}^L - z_{Q10}^R \end{aligned} \tag{13}$$

and $r_{Q10}^R = \left[x_{Q10}^R\ ,\ y_{Q10}^R\ ,\ z_{Q10}^R\right]^T$ is the vector from the origin 0 of inertial frame R to the origin Q_1 of actuator fixed frame K_1. The transformation of the position and orientation vector x_1 to the velocity vector x_2 (comp (4)) is defined by the relation

$$\frac{d\,x_1}{dt} = T(x_1)\ x_2 \tag{14}$$

which includes the nonlinear Poisson equations, where

$$T(x_1) = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{bmatrix} = \begin{bmatrix} 1\ ,\ 0\ ,\ 0\ , & 0\ , & 0\ , & 0 \\ 0\ ,\ 1\ ,\ 0\ , & 0\ , & 0\ , & 0 \\ 0\ ,\ 0\ ,\ 1\ , & 0\ , & 0\ , & 0 \\ 0\ ,\ 0\ ,\ 0\ , & c_3/c_2\ , & -s_3/c_2\ , & 0 \\ 0\ ,\ 0\ ,\ 0\ , & s_3\ , & c_3\ , & 0 \\ 0\ ,\ 0\ ,\ 0\ , & -c_3\ t_2\ , & s_3\ t_2\ , & 1 \end{bmatrix} \in R^{6\,6}, \quad (t_2 = \tan(\theta)) \tag{15}$$

2.2 SERVOHYDRAULIC ACTUATORS

The servohydraulic actuators are modeled by the equations

$$\dot{p} = C_H^{-1} \left[Q_x \; K_V \; u + Q_P \; p - A_k \; J_{x1}(x_1) \; x_2 \right] \; \in \mathbb{R}^6 \tag{16}$$

where u is the vector of servovalve command inputs

$$u = \left[u_1, \; u_2, \; u_3, \; u_4, \; u_5, \; u_6 \right]^T \; \in \mathbb{R}^6 \tag{17}$$

and

$$C_H = diag\left(C_{H1}, \; \ldots, \; C_{H6}\right), \quad Q_p = diag\left(Q_{p1}, \; \ldots, \; Q_{p6}\right), \tag{18}$$

$$Q_x = diag\left(Q_{x1}, \; \ldots, \; Q_{x6}\right), \quad K_V = diag\left(K_{V1}, \; \ldots, \; K_{V6}\right) \tag{19}$$

are the diagonal matrices of the actuator hydraulic capacities, servovalve pressure coefficients, servovalve displacement coefficients and servovalve gain factors, respectively. The servovalve mechanics model is omitted in this approach.

2.3 NONLINEAR STATE SPACE REPRESENTATION OF THE PLANT

The combined Equations (1) and (16) may be written in nonlinear state space form

$$\dot{x} = f(x) + B \; u = f(x) + \sum_{j=1}^{6} b_j \, u_j \quad ; \quad x \in \mathbb{R}^{18} \quad , \quad u \in \mathbb{R}^6 \tag{20}$$

where

$$x = \begin{bmatrix} p \\ x_1 \\ x_2 \end{bmatrix} \in \mathbb{R}^{18}, \quad B = \left[b_1, \; b_2, \; b_3, \; b_4, \; b_5, \; b_6 \right] = \begin{bmatrix} B_1 \\ 0_{12\,6} \end{bmatrix} \in \mathbb{R}^{18,6} \tag{21}$$

$$B_1 = \begin{bmatrix} b_{11}, & 0, & 0, & 0, & 0, & 0 \\ 0, & b_{22}, & 0, & 0, & 0, & 0 \\ 0, & 0, & b_{33}, & 0, & 0, & 0 \\ 0, & 0, & 0, & b_{44}, & 0, & 0 \\ 0, & 0, & 0, & 0, & b_{55}, & 0 \\ 0, & 0, & 0, & 0, & 0, & b_{66} \end{bmatrix} = C_H^{-1} \; Q_x \; K_V \; \in \mathbb{R}^{6\,6} \tag{22}$$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} C_H^{-1} \; Q_p \; p - C_H^{-1} \; A_K \; J_{x1}(x_1) \; x_2 \\ T(x_1) \; x_2 \\ \tilde{\alpha}(x_1, x_2) + \tilde{\beta}(x_1) \; p \end{bmatrix} \in \mathbb{R}^{18} \tag{23}$$

$$\tilde{\alpha}(x_1, x_2) = M^{-1}(x_1) \left\{ -J_{x1}^T(x_1) \; D_k \; J_{x1}(x_1) \; x_2 - n(x_1, x_2) - q_G(x_1) \right\} \; \in \mathbb{R}^6, \tag{24}$$

$$\tilde{\beta}(x_1) = M^{-1}(x_1)\, J_{x1}^{T}(x_1) \cdot A_k \quad \in \mathbb{R}^6 \tag{25}$$

Clearly, $f(x)$ and B in (20) are smooth vector fields

3. Nonlinear Controller Design

In all known control theoretical investigations of multi-axis test facilities from industrial practice the nonlinear equations of motion are linearized locally (compare Skelton, 1988, Hahn, 1986). The control concept investigated in this paper is based on exact linearization of the nonlinear control plant using nonlinear feedback compensation controllers and a linear state feedback controller. The design steps to be performed may be summarised as follows:

- Test of exact linearization conditions
- Construction of a nonlinear state transformation (diffeomorphism)
- Mapping of nonlinear control plant equations into nonlinear normal form
- Nonlinear feedback controller design
  - feedback compensation of plant nonlinearities (exact linearization)
  - pole placement of the exact linearized control plant via linear state feedback (stabilisation).

3.1 TEST OF EXACT LINEARIZATION CONDITIONS

The nonlinear control plant (20) is for rank $B_1=6$ according to a Lemma of Isidori exactly linearizable, if and only if smooth formal output functions

$$y_i = h_i(x) = h_i(p, x_1, x_2) \in R^1, \quad (i = 1, \ldots, 6) \tag{26}$$

exist on $U(x_0)$which satisfy the following conditions (Isidori, 1989; Schwarz 1991, Slotine 1991)

-1 $L_{b_j} L_f^k h_i(x) = 0$ (27)
for $1 \le i, j \le 6$, for $k < r_i - 1$ and for all x of a neighbourhood $U(x_0)$ of x_0.

-2 $E(x) = \begin{bmatrix} L_{b_1} L_f^{r_1-1} h_1(x), & \ldots, & L_{b_6} L_f^{r_1-1} h_1(x) \\ \cdot & , & \cdot \\ L_{b_1} L_f^{r_6-1} h_6(x), & \ldots, & L_{b_6} L_f^{r_6-1} h_6(x) \end{bmatrix}$ is non-singular at $x = x_0$ and (28)

-3 $\sum_{i=1}^{6} r_i = n$ (29)

These relations define the vector relative degree $(r_1, r_2, r_3, r_4, r_5, r_6)$ of the system (20) together with (26), where $L_f^k h_i(x)$ are Lie derivatives of $h_i(x)$ with respect to f The general

formal output functions (26) have been specialised in three steps

-1 Choice of $h(p,x_1,x_2)$ independent of p implies

$$L_{b_j} L_f^0 h_i(x_1,x_2) = \nabla h(x)\; b_j \equiv 0 \;,\; i,j = 1, \quad ,6 \text{ and } r_i \geq 1 \quad (i = 1, \quad ,6) \tag{30}$$

-2 Choice of $h(x_1,x_2)$ independent of x_2 implies

$$L_{b_j} L_f^1 h_i(x_1) \equiv 0 \;,\; i,j = 1, \quad ,6 \quad \text{and} \quad r_i \geq 2 \quad (i = 1, \quad ,6) \tag{31}$$

-3. If $h(x_1)$ is selected as

$$y = h(x_1) := x_1 \,, \tag{32}$$

condition

$$L_{b_j} L_f^2 h_i(x_1) \not\equiv 0 \;;\; i = 1, \quad ,6 \text{ implies } r_i = 3 \;(i = 1, \quad ,6) \text{ and } \sum_{i=1}^{6} r_i = 18 \;, \tag{33}$$

where

$$-1 \quad L_f^1 h_i(x_1) = f_{2i} = t_i(x_1)\; x_2 \;,\; i = 1, \quad ,6 \tag{34}$$

$$-2. \quad L_f^2 h_i(x_1) = \hat{t}_i(x_1,x_2)\; T(x_1)\; x_2 + t_i(x_1) \cdot f_{3i} \;,\; i = 1, \;. \quad ,6 \tag{35}$$

where in agreement with (23)

$$f_3(x) = \begin{bmatrix} f_{31} \\ \cdot \\ f_{36} \end{bmatrix} = \tilde{\alpha}(x_1,x_2) + \tilde{\beta}(x_1)\; p \quad \in \mathbb{R}^6 \,, \tag{36}$$

$$\hat{T}(x_1,x_2) = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \\ \hat{t}_3 \\ \hat{t}_4 \\ \hat{t}_5 \\ \hat{t}_6 \end{bmatrix} = \begin{bmatrix} 0 \;,0 \;,0 \;,0 \;, & 0 & , & 0 \\ 0 \;,0 \;,0 \;,0 \;, & 0 & , & 0 \\ 0 \;,0 \;,0 \;,0 \;, & 0 & , & 0 \\ 0 \;,0 \;,0 \;,0 \;, & \frac{c_3\; s_3}{c_2^2}\, p^L - \frac{s_3\; s_2}{c_2^2}\, q^L & , & -\frac{s_3}{c_2}\, p^L - \frac{c_3}{c_2}\, q^L \\ 0 \;,0 \;,0 \;,0 \;, & 0 & , & c_3\; p^L - s_3\; q^L \\ 0 \;,0 \;,0 \;,0 \;, & -\frac{c_3}{c_2^2}\, p^L + \frac{s_3}{c_2^2}\, q^L & , & s_3\; t_2\; p^L + c_3\; t_2\; q^L \end{bmatrix} \tag{37}$$

and (38)

$$-3 \quad \left[L_{b_j} L_f^2\; h_i(x_1) \right]_{\substack{i=1 \quad ,6 \\ j=1 \quad ,6}} = \left[t_i\, \frac{\partial f_{3i}}{\partial p_j}\, b_{jj} \right]_{\substack{i=1, \quad 6 \\ j=1, \quad 6}} = E(x) = \begin{bmatrix} L_{b_1} L_f^2\; h_1(x_1) \,, & , & L_{b_6} L_f^2\; h_1(x_1) \\ & , & \\ L_{b_1} L_f^2\; h_6(x_1) \,, & , & L_{b_6} L_f^2\; h_6(x_1) \end{bmatrix}$$

Then using (25) yields

$$E(x) = T(x_1)\, \frac{\partial f_3}{\partial p} \cdot B_1 = T(x_1)\; \tilde{\beta}(x_1)\; B_1 = T(x_1)\; M^{-1}(x_1)\; J_{x1}^T(x_1)\; A_K\; B_1 \tag{39}$$

Matrix T (compare (15)) is regular for values of $\theta \neq \pi/2 + n\,\pi$ $(n = 0, 1, \ldots)$ The inertia matrix $M(x_1)$ is always regular (compare (2)) The diagonal matrices A_k and B_1 are regular, too, for a test facility driven by six actuators (compare (3) and (25)) Then $E(x)$ is regular (excluding the critical values of θ) if and only if $J_{x1}(x_1)$ is a regular matrix. There exist pathological actuator attachment configurations where $J_{x1}(x_1)$ may become locally singular (compare Figure 4) Those configurations and isolated locations excluded $J_{x1}(x_1)$ is regular and

$$\det\left(J_{x1}(x_1)\right) \neq 0 \tag{40}$$

Then $E(x)$ is regular, too, and the control plant (20) is exactly linearizable by state feedback

3.2 CONSTRUCTION OF THE NONLINEAR STATE TRANSFORMATION

Based on the previous design steps, the nonlinear diffeomorphism which maps (20) into nonlinear normal form is defined by the relation

$$\Phi : \mathbb{R}^{18} \to \mathbb{R}^{18}, \quad \begin{bmatrix} p \\ x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

where (using (32), (34) and (35)) (41)

$$\underbrace{\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}}_{=z} = \underbrace{\begin{bmatrix} \Phi_1(x) \\ \Phi_2(x) \\ \Phi_3(x) \end{bmatrix}}_{=\Phi(x)} = \begin{bmatrix} x_1 \\ T\,x_2 \\ \dot{T}\,\dot{T}\,x_2 + T\,f_3 \end{bmatrix} = \begin{bmatrix} x_1 \\ T\,x_2 \\ \dot{T}\,T\,x_2 + T\,\tilde{\alpha}(x_1, x_2) + T\,\tilde{\beta}(x_1)\,p \end{bmatrix}, \quad z_i \in \mathbb{R}^6, \ (i = 1, \ldots, 6)$$

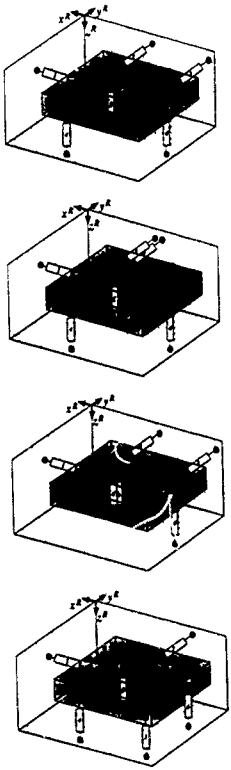
and Φ is a smooth vector field

Note: (Physical interpretation of normal form state vector z)

Taking into account (15) the normal form state vector $z = \left[z_1^T, z_2^T, z_3^T\right]^T \in \mathbb{R}^{18}$ has the following pleasant physical interpretation:

- z_1 is identical to the test table position and orientation vector $x = \left[x_P^R, y_P^R, z_P^R, \varphi, \theta, \psi\right]^T$
- z_2 is identical to the test table velocity and angular velocity vector $x_1 = T(x_1)\,x_2$ and
- z_3 is identical to the test table acceleration and angular acceleration vector (compare (41))

Due to (40) the smooth inverse vector field $\Phi^{-1}(z)$ exists and is described by the expression



regular case $T_d = \begin{bmatrix} 1, 0\ 0 & 0 & z_{P1P} & ,-y_{P1P} \\ 0, 1\ 0, & -z_{P2P}, & 0 & x_{P2P} \\ 0, 1, 0, & -z_{P3P}, & 0 & , x_{P3P} \\ 0, 0, 1, & y_{P4P} & ,-x_{P4P}, & 0 \\ 0, 0, 1, & y_{P5P} & ,-x_{P5P}, & 0 \\ 0, 0, 1 & y_{P6P} & ,-x_{P6P}, & 0 \end{bmatrix}$

singular case $T_d = \begin{bmatrix} 1, 0, 0 & 0 & z_{P1P} & ,-y_{P1P} \\ 0, 1, 0, & -z_{P2P}, & 0 & x_{P2P} \\ 0, 1, 0, & -z_{P2P} & 0 & , x_{P2P} \\ 0, 0, 1, & y_{P4P} & -x_{P4P}, & 0 \\ 0, 0, 1, & y_{P5P} & -x_{P5P}, & 0 \\ 0, 0, 1, & y_{P6P} & ,-x_{P6P}, & 0 \end{bmatrix}$

singular case $T_d = \begin{bmatrix} 1, 0, 0 & 0 & , z_{P1P} & ,-y_{P1P} \\ 0, 1\ 0, & -z_{P2P}, & 0 & , x_{P2P} \\ 0, 1, 0 & -z_{P3P}, & 0 & , x_{P3P} \\ 0, 0, 1 & y_{P4P} & -x_{P4P}, & 0 \\ 0, 0\ 1, & y_{P4P} & -x_{P4P}, & 0 \\ 0, 0\ 1 & y_{P6P} & -x_{P6P} & 0 \end{bmatrix}$

singular case $T_d = \begin{bmatrix} 1\ 0\ 0 & 0 & z_{P1P} & -y_{P1P} \\ 0\ 1\ 0 & -z_{P2P} & 0 & x_{P2P} \\ 0, 1\ 0 & y_{P3P} & -x_{P3P} & 0 \\ 0\ 0\ 1 & y_{P4P} & -x_{P4P} & 0 \\ 0\ 0\ 1 & y_{P5P} & ,-x_{P5P}, & 0 \\ 0, 0\ 1 & y_{P6P} & -x_{P6P} & 0 \end{bmatrix}$

Figure 4 Singular cases of the mapping $J_{x1}(x_1)$ for small values of x_1

$$z = \begin{bmatrix} p \\ x_1 \\ x_2 \end{bmatrix} = \Phi^{-1}(z) = \begin{bmatrix} \tilde{\beta}^{-1}(z_1)\, T^{-1} \left\{ z_3 - \hat{T}\, z_2 - T\, \tilde{\alpha}(z_1, z_2) \right\} \\ z_1 \\ T^{-1}\, z_2 \end{bmatrix} \in I\!R^{18} \qquad (42)$$

3.3 DERIVATION OF NORMAL FORM CONTROL PLANT EQUATIONS

Using the nonlinear coordinate transformation (41) and its time derivative

$$\dot{z} = \frac{\partial \Phi}{\partial x} \cdot \dot{x} \qquad (43)$$

the nonlinear control plant equations (20) are mapped to nonlinear normal form

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ \dot{z}_3 &= \alpha(z_1, z_2, z_3) + \beta(z_1)\, u \end{aligned} \qquad (44)$$

where

$$\Phi_3 = \hat{T}(x_1, x_2)\, T(x_1)\, x_2 + T(x_1)\, \tilde{\alpha}(x_1, x_2) + T(x_1)\, \tilde{\beta}(x_1)\, p \,,$$

$$\alpha(z_1, z_2, z_3) = \left[\frac{\partial \Phi_3}{\partial p} \cdot f_1(x) + \frac{\partial \Phi_3}{\partial x_1} \cdot f_2(x) + \frac{\partial \Phi_3}{\partial x_2} \cdot f_3(x) \right]_{x = \Phi^{-1}(z)} \quad \text{or} \qquad (45a)$$

$$\begin{aligned} \alpha(z_1, z_2, z_3) = {} & \left[M^{-1}(z_1)\, J_{x_1}^T(z_1) \cdot A_k\, f_1(x) + \right. \\ & \frac{\partial}{\partial x_1} \left[\hat{T}(x_1, x_2)\, T(x_1)\, x_2 + T(x_1)\, \tilde{\alpha}(x_1, x_2) + T(x_1)\, \tilde{\beta}(x_1)\, p \right] f_2(x) + \qquad (45b) \\ & \left. \frac{\partial}{\partial x_2} \left[\hat{T}(x_1, x_2)\, T(x_1)\, x_2 + T(x_1)\, \tilde{\alpha}(x_1, x_2) \right] f_3(x) \right]_{x = \Phi^{-1}(z)} \end{aligned}$$

and

$$\beta(z_1) = \left. \frac{\partial \Phi_3}{\partial p} \right|_{x = \Phi^{-1}(z)} \mathrm{B}_1 = T(z_1)\, M^{-1}(z_1)\, J_{z_1}^T(z_1)\, A_k\, \mathrm{B}_1 \qquad (45c)$$

3.4 NONLINEAR CONTROLLER DESIGN

After having derived the nonlinear normal form control plant equations the controller design is straight forward and includes the following standard design steps (compare Isidori, 1989, Slotine, 1991, Schwarz 1991)

- feedback compensation of the plant nonlinearities (exact linearization) and
- pole and zero placement of the overall control loop via linear state feedback (stabilisation and transient shaping)

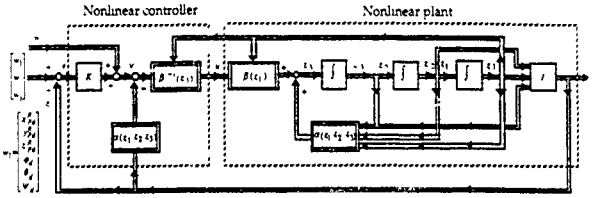


Figure 5 Block diagram of the overall control system

Implementation of a nonlinear feedback control law

$$u = \beta^{-1}(z_1)\left[v - \alpha(z_1, z_2, z_3)\right] \in R^6 \tag{46}$$

according to Figure 5 yields the system equations among input vector v and system state z

$$\dot{z}_1 = z_2 \quad , \quad \dot{z}_2 = z_3 \quad , \quad \dot{z}_3 = v \tag{47}$$

The poles and zeros of this (degenerate) system are arbitrarily placed by the linear state feedback law (48) including a third order time derivative w_1 of the input vector w_1 (for zero placement and transient shaping).

$$v = w_1 - K\ (z - w) \quad \in R^6 \tag{48}$$

The resulting overall control (cf. 5) system is represented by the equations

$$\dot{z}_1 = z_2 \quad , \quad \dot{z}_2 = z_3 \quad , \quad \dot{z}_3 = w_1 + K\,(z - w) = 0 \tag{49}$$

where

$$K = \left[K_1\ ,\ K_2\ ,\ K_3\right] \ ,\ K_i = diag\ \left(k_{i1}\ ,\ k_{i2}\ ,\ k_{i3}\ ,\ k_{i4}\ ,\ k_{i5}\ ,\ k_{i6}\right)\ ,\ i = 1\ ,\quad ,\ 6$$

and

$$\begin{aligned} w &= \left[w^T_1\ ,\ w^T_2\ ,\ w^T_3\right]^T \text{(command input)}, & z &= \left[z_1^T\ ,\ z_2^T\ ,\ z_3^T\right]^T \text{(state vector)}\ , \\ w_1 &= \left[x^R_{Pd}\ ,\ y^R_{Pd}\ ,\ z^R_{Pd}\ ,\ \varphi_d\ ,\ \theta_d\ ,\ \psi_d\right]^T, & z_1 &= \left[x^R_P\ ,\ y^R_P\ ,\ z^R_P\ ,\ \varphi\ ,\ \theta\ ,\ \psi\right]^T, \\ w_2 &= \left[\dot{x}^R_{Pd}\ ,\ \dot{y}^R_{Pd}\ ,\ \dot{z}^R_{Pd}\ ,\ \dot{\varphi}_d\ ,\ \dot{\theta}_d\ ,\ \dot{\psi}_d\right]^T, & z_2 &= \left[\dot{x}^R_P\ ,\ \dot{y}^R_P\ ,\ \dot{z}^R_P\ ,\ \dot{\varphi}\ ,\ \dot{\theta}\ ,\ \dot{\psi}\right]^T, \\ w_3 &= \left[\ddot{x}^R_{Pd}\ ,\ \ddot{y}^R_{Pd}\ ,\ \ddot{z}^R_{Pd}\ ,\ \ddot{\varphi}_d\ ,\ \ddot{\theta}_d\ ,\ \ddot{\psi}_d\right]^T & z_3 &= \left[\ddot{x}^R_P\ \ \ddot{y}^R_P\ \ \ddot{z}^R_P\ \ \ddot{\varphi}\ ,\ \ddot{\theta}\ ,\ \ddot{\psi}\right]^T \end{aligned} \tag{50}$$

4. Computer Simulation Results

The nonlinear control system of Figure 5 with nonlinear control plant (20) and with nonlinear controller (46) has been investigated in a computer simulation using standard transient test signals w_1 for vibration testing of space structures (as shown in Figures 6a, 7a, 8a and 9a) together with their first, second and third time derivatives as command input vector w. A complete and ideal nonlinear control law (46) will exactly reproduce those input signals as output vector of the control loop. In industrial practice not all plant parameters are known exactly. As a consequence, the plant parameter values used in the control law may deviate from the parameter values realised in the plant. The influence of those parametric discrepancies between control plant and controllers is investigated subsequently by computer simulations. Each of the following variations of controller parameters is counted with respect to a set of nominal values of controller parameters which are identical to the plant parameters. Variations of the **hydraulic capacity** C_H which stand for variations of the oil compressibility due to temperature variations and due to oil contamination's caused by air bubbles are shown in Figure 6. The tracking behaviour is only influenced slightly by variations of C_H of 50 % (cf. 6b and 6c). Unrealistic variations of C_H by a factor four or more modify the tracking behaviour significantly according to Figure 6d. The coupling of the mechanical degrees of freedom is insensitive with respect to those parameter variations. In Figures 7b and 7c the common **mass m of the test table and payload** is modified by a factor 1/2 and 2, respectively. This drastic mass variation again does not severely influence both, the tracking and the coupling behaviour of the system. Only unrealistic variations of m by a factor four or more have significant influence on the system behaviour (cf. 7d). Variations of all **moments and products of inertia** of the same order have no significant effect on the system behaviour. These results are omitted here. Variations of the common **center of gravity** of the test facility and payload are shown in Figure 8. Simultaneous huge modifications of all components of the vector r_{CP} from reference point P to the center of gravity C (for fixed P) by factors 1/2 and 2 are shown in Figures 8b and 8c. They don't affect the tracking behaviour and only provide slight couplings among the test table degrees of freedom. Only unrealistic variations of r_{CP} by a factor four or more show significant modifications of the coupling behaviour of the system (cf. 8d). The nominal plant model equations (20) did not include nonlinear **friction forces** due to dry friction in the hydraulic actuators and in the joints attached to those. As a consequence those friction forces have not been taken into account in the nonlinear control law design. Figure 9 shows simulation results including nonlinear friction forces of different absolute values in the plant model. Friction forces F_R smaller than 3 KN don't provide unwanted influences upon the system behaviour (cf. 9a). Friction forces of the order of F_R = 5 KN don't affect the tracking behaviour. They introduce small coupling effects of some test facility degrees of freedom (cf. 9b). Friction forces of the order of F_R = 10 KN introduce slight modifications of the tracking behaviour and spontaneously occurring coupling of some degrees of freedom. Hugh friction forces of the order of F_R = 20 KN which are unrealistic in the applications considered severely deteriorate both, the tracking and the decoupling behaviour of the test facility. Those extreme friction forces are avoided by using actuators with hydrostatic bearings in the test facility considered. The above numerical sensitivity analysis of the control loop considered demonstrates that the exact linearization approach is much less sensitive to plant parameter variations and to controller parameter variations than often predicted. As a consequence various processes and machines from industrial practice can be efficiently controlled by using this approach.

5 Discussion of Possible Practical Limitations of the Control Concept

In many nonlinear control applications from industrial practice, exact linearization techniques is of minor importance due to the following limitations

- (1) the theoretical and numerical amount of work
  - to test the exact linearization conditions,
  - to calculate both, a suitable diffeomorphism and its inverse and
  - to calculate the normal form control plant equations and the nonlinear controllers is extreme cumbersome and expensive and is often beyond the limits of practical feasibility,
- (2) the plant model equations are not accurately known,
- (3) the plant model data are not exactly known,
- (4) the full state vector is not available for feedback purposes,
- (5) the time derivatives of the command input signals are not available up to a sufficiently high order,
- (6) the lengthy and complex control laws cannot implemented as on-line algorithms.

In the previously discussed control of a spatial multi-axis servohydraulic test facility these practical limitations have been overcome as follows

- (1) the test of exact linearization conditions and the calculation of a suitable diffeomorphism, of its extreme lengthy inverse, of the nonlinear normal form control plant representation and of the controller algorithms have been done by systematically using symbolic languages (some of these formal expressions have typewritten length of several thousand DIN A4 pages, compare Table 1, column 1),
- (2) the plant model equations (including the test table and payload mechanics and the actuator hydraulics) are well known with sufficient accuracy,
- (3) the associated model data are well known both, from numerical calculations and from laboratory experiments; in addition, simulation results prove that the control concept proposed is comparatively insensitive with respect to variations of relevant plant parameters The only essential model data that are not well known in the above application are dry friction forces in the actuators They may be drastically reduced or even avoided by using actuators with hydrostatic bearings;
- (4) all 18 coordinates of the state vector z of the normal form control plant equations (including test table positions and orientations and the associated velocities and accelerations) are available as sensor outputs;
- (5) the command input vector w_1 as well as its first, second and third derivatives w_2 , w_3 and w_3 are available as test signals before starting a test experiment,
- (6) the extreme lengthy and time consuming control algorithms have been drastically reduced by suitable substitutions of repetitively occurring internal expressions(compare Table 1, column 2), they can be implemented as real time algorithms in transputers or signal processors

	number of pages of symbolic expression	number of pages of implemented expression
α	9614	43
β	1630	63
Φ^{-1}	1570	55

Table 1. Length of symbolic expressions and implemented algorithms of the nonlinear control laws (counted in typewritten DIN A4 pages) computed by using the symbolic language MACSYMA

6. References

Isidori, A (1989) Nonlinear Control Systems An Introduction. Springer Verlag, pp. 234-288

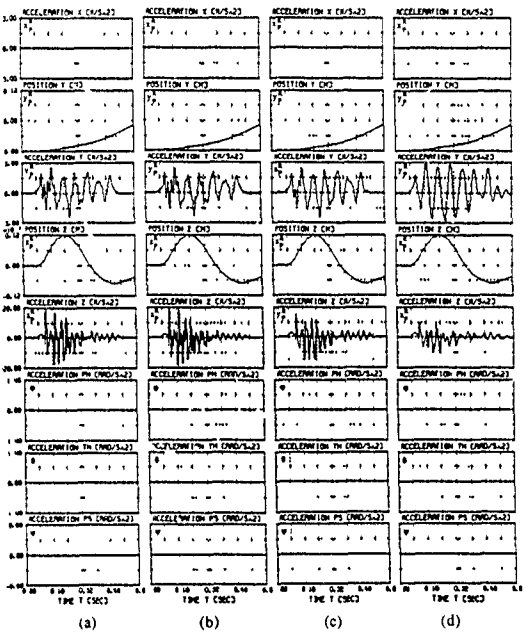
Hahn, H., Leimbach, K.-D and Zhang, X. (1992). Theoretical Modeling and Control Concept of a Linear Multi-Axis Servohydraulic Test Facility. IFAC Symposium LSS' 92, Peking

Skelton, R E (1988). Dynamic Systems Control. J Wiley.

Hahn, H and Raasch, W (1986). Multi-Axis Vibration Test On Spacecraft Using Hydraulic Exciters AGARD Conference Proceedings No.397, pp 23-1 to 23-23

Schwarz, H (1991). Nichtlineare Regelungssysteme Oldenbourg Verlag

Slotine, J J. and Li, W. (1991). Applied Nonlinear Control Prentice Hall, pp. 207-271



(a) (b) (c) (d)

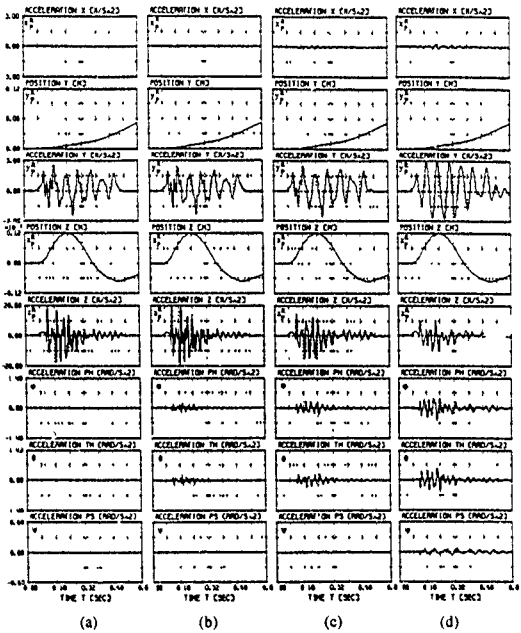
Figure 6 Time histories of the test table degrees of freedom obtained from computer simulations including parameter variations of the actuator hydraulic capacities C_H

(a) Nominal value of C_H

(b) Variation of C_H by a factor of 0 5

(c) Variation of C_H by a factor of 2 0

(d) Variation of C_H by a factor of 4 0



(a) (b) (c) (d)

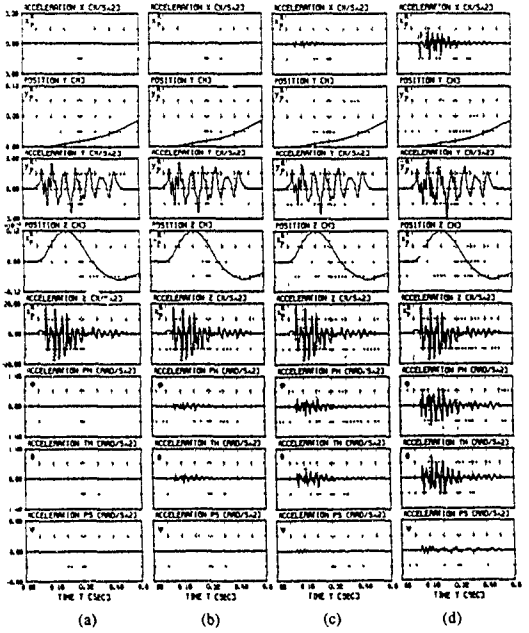
Figure 7 Time histories of the test table degrees of freedom obtained from computer simulations including parameter variations of the common mass m of test table and payload

(a) Nominal value of m

(b) Variation of m by a factor of 0.5

(c) Variation of m by a factor of 2 0

(d) Variation of m by a factor of 4 0



(a) (b) (c) (d)

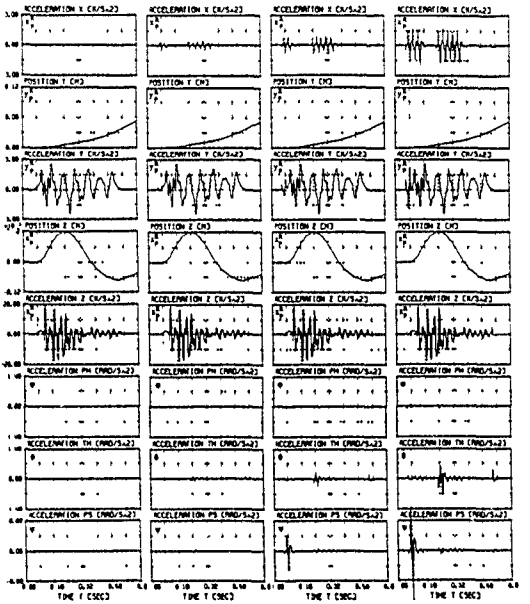
Figure 8 Time histories of the test table degrees of freedom obtained from computer simulations including parameter variations of the common center of gravity

(a) Nominal value of r_{CP}

(b) Variation of r_{CP} by a factor of 0 5

(c) Variation of r_{CP} by a factor of 2 0

(d) Variation of r_{CP} by a factor of 4 0



(a) (b) (c) (d)

Figure 9 Time histories of the test table degrees of freedom obtained from computer simulations including dry friction forces within the servohydraulic actuators and the joints

(a) Actuators and joints without friction

(b) Friction force of F_R = 5 KN

(c) Friction force of F_R = 10 KN

(d) Unrealistic high friction force of F_R = 20 KN

IDENTIFICATION OF MINIMUM SET PARAMETERS OF FLEXIBLE ROBOTS

P. CHEDMAIL, Ph. DEPINCE, F BENNIS

Laboratoire d'Automatique de Nantes UA C.N.R.S. n° 823
Ecole Centrale de Nantes et Université de Nantes
1 rue de la Noé, 44072 Nantes Cedex 03, France
Tél. : 40 37 16 00 Fax : 40 74 74 06.

ABSTRACT

The object of this paper is to present new results about the modelling and identification of flexible robots. The existence of a minimum set of parameters is proved by the use of a symbolic method. The determination of this minimum set increases the robustness of the identification process.
The identification model based on the energy theorem is shown to be linear in terms of a set of physical parameters called standard parameters. A necessary and sufficient condition which ensures the minimality of the standard parameters is given. This condition depends on eigen functions if the model derives from an assumed modes method and on shape functions if the model derives from a finite elements method.
In the case of robots whose links are all flexible, we demonstrate that the only possible regroupings of parameters are obtained with parameters belonging to the same link.

1. Introduction

The active control of robot flexibilities requires considerable effort in terms of modelling and identification. Compared with rigid robots, the degree of difficulty in the case of flexible robots is increased because of the existence of non linear interactions between the large rigid displacements and flexible components of the elastic deformations. We can find in [1] and [2] a summary of different methods The main differences are :

- the choice of parameters - in particular, the choice of reference frames for the description of elastic displacements-,
- the choice of the approximation of the deflection - the most common approximate description are based on assumed modes method or finite elements method-,
- the physical principle used to obtain the dynamic model - virtual work principle or Lagrange's equations -,
- the way these equations are calculated -the use of a symbolic or a numerical method-,
- the degree of approximation -development order of the model, order of magnitude of the elastic displacements,...-.

The control of such structures has been extensively studied. The main approaches are : methods of non exciting trajectories [3], adaptative control [4], control using a knowledge model. In this last case, we can find singular perturbation methods [5], LQR methods based on an approximated linearized model [6] or non linear decoupling methods [7].

Our present work is based on the knowledge model approach. We have noticed in previous works [8, 9] that control laws are prone to errors made in the evaluation of the parameters which describe the model. It is therefore necessary to find a compromise between the complexity of the model and its time computation. A first stage consists in a structural reduction of the model -the choice of the number of finite elements or eigen functions describing the flexibilities of the robot determines its dynamical performances A second stage allows us to find the best parameters - and to identify them - in order to approximate the real behaviour of the robot - even if the chosen model is not accurate - and to make its computation possible.

The identification of the parameters of rigid robots has been studied in [10,...,14] On the one hand, identification ensures the on-line determination of the parameters of the robot for control purpose. On the other hand, the determination of the set of minimum parameters of robots reduces the computational cost of dynamic model and increases the robustness of the identification process.

In the case of flexible robots, several identification schemes have been studied. Some on-line identification schemes are based on input-output ARMA representations [15, 16, 17]. Another approach consists in elaborating a minimal identification model based on a knowledge model of the robot and in applying a least-squares method. There are two kinds of minimal identification model : the first consists in applying the theorem of energy for the robot, the second comes from the dynamic model. More details on these two models applied in the case of one or two link-planar robots can be found in [18] and [19]. A set of standard parameters has been proposed. Its minimality has been demonstrated using a numerical rank analysis of the observation matrix which is constructed with a random sequence of points.

In this paper we offer a new direct formal analysis of the minimality of the standard parameters. The paper is organized as follows :

- The first part deals with the modelling of the flexible robots and the determination of a standard set of parameters The identification model is deduced from a physical equating of the theorem of energy. We demonstrate that the identification model can be written in a linear form with respect to the standard set of parameters.
- A second part is devoted to the problem of the minimality of the standard set of parameters. We give a necessary and sufficient condition which is related to the choice of the eigen (or shape) functions of the discretization.

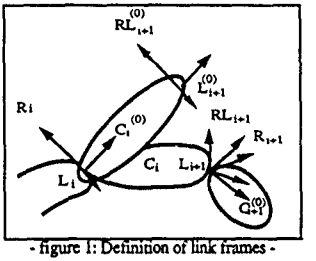
2. Modelling Of Flexible Robots

We study thereafter a single chain open-loop robot structure. The elastic deformations of each link with respect to its rigid configuration are supposed to be small.
The elastic motions of each link are referred to the position of the undeformed corresponding link.

2.1. DESCRIPTION OF THE ROBOT

The different frames used to define the model of the flexible robot are described on figure 1.

- $C_i^{(0)}$ is the rigid configuration of link i,
- C_i is the deformed configuration of link i,
- L_{i+1} is the center of the joint between link i and link i+1,
- $R_i = (L_i, X_i, Y_i, Z_i)$ is a frame attached to link C_i,
- $RL_{i+1}^{(0)} = (L_{i+1}, X_i, Y_i, Z_i)$ is a frame, parallel to frame R_i and centred on $L_{i+1}^{(0)}$,
- $RL_{i+1} = (L_{i+1}, X_i', Y_i', Z_{i+1})$ is a transformed frame from $RL_{i+1}^{(0)}$ after the elastic deformation of $C_i^{(0)}$,
- $R_{i+1} = (L_{i+1}, X_{i+1}, Y_{i+1}, Z_{i+1})$ is a transformed frame from RL_{i+1} in the transformation function of the joint variable q_{ri+1}.



- figure 1: Definition of link frames -

2.2. TRANSFORMATION MATRICES

All the transformations are represented with the homogeneous transformation matrices The global transformation matrix from R_i to R_{i+1} is defined as follows:

$$R_iT_{Ri+1} = R_iT_{RL_{i+1}}^{(0)}\, E(L_{i+1})\, {}^{RL_{i+1}}T_{R_{i+1}} \qquad (1)$$

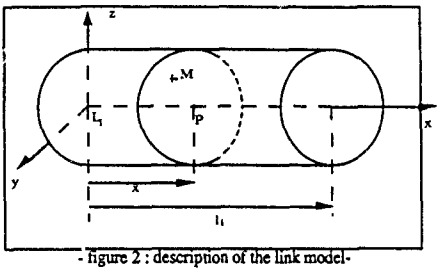
where

- ${}^{R_i}T_{RL_{i+1}}^{(0)}$ is a given constant transformation from R_i to $RL_{i+1}^{(0)}$: translation of vector $L_iL_{i+1}^{(0)}$.
- $E(L_{i+1}) = {}^{RL_{i+1}^{(0)}}T_{RL_{i+1}}$ is the elastic transformation matrix from $RL_{i+1}^{(0)}$ to RL_{i+1}. Its components are a function of the elastic displacement and rotation of L_{i+1} of C_{i+1} with respect to its rigid configuration $L_{i+1}^{(0)}$.
- ${}^{RL_{i+1}}T_{Ri+1}$ is the transformation matrix from RL_{i+1} to R_{i+1}. It's a function of the rigid joint variable q_{Ri+1}.

2.3. DEFINITION OF THE ELASTIC DEFORMATIONS

We now consider that each link is modelized with a deformable beam. Moreover, we suppose that the elastic displacement field of each beam verifies Euler-Bernoulli's hypothesis.

Let M be a point of the cross-section of the beam which is perpendicular to the neutral axis. Let P be the point of the neutral axis which intersects the same cross section -Figure 2. The position of any point M - belonging to link C_i - with respect to frame R_i is obtained by superposing the elastic displacement due to the link flexibility ($U_e(M)$) to the rigid position of M tied to $\overset{(0)}{C_i}$ ($U_r(M)$).



- figure 2 : description of the link model-

$$R_iL_iM = R_iU_r(M) + R_iU_e(M) \quad (2)$$

where :

- $R_iU_r(M) = [x, y, z]^T$
- $R_iU_e(M) = R_iU_e(P) + R_i\theta_e(P) \wedge R_iPM$ (3)
- $R_iPM = [0, y, z]^T$

The elastic displacement vector of the neutral axis, $[R_iU_e(P)^T, R_i\theta_e(P)^T]^T$, can be defined with respect to the modal or nodal coordinates depending on the discretization : modal expansion or finite elements method.
If a modal expansion method is used, the expression of the elastic displacement vector of the neutral axis is given as follows :

$$\begin{bmatrix} R_iU_e(P) \\ R_i\theta_e(P) \end{bmatrix} = \sum_{j=1}^{m_i} \underline{\Phi}_j^i(x)\, q_j^i(t) \quad (4)$$

with :

$\underline{\Phi}_j^i(x) = [\Phi_{jux}^i(x), \Phi_{juy}^i(x), \Phi_{juz}^i(x), \Phi_{j\theta x}^i(x), \Phi_{j\theta y}^i(x), \Phi_{j\theta z}^i(x)]^T$ is the eigen functions vector.

If a finite element method is used, the expression of the elastic displacement vector of the neutral axis is given as follows:

$$\begin{bmatrix} R_iU_e(P) \\ R_i\theta_e(P) \end{bmatrix} = N^i(x).{}^{R_i}U_{ei} \quad (5)$$

where :

- ${}^{R_i}U_{ei}$ is the (12x1) vector of nodal coordinates :

${}^{R_i}U_{ei} = [ux_1^i, uy_1^i, ..., \theta z_1^i, ux_2^i, uy_2^i, \quad , \theta z_2^i]^T$

- $N^i(x)$ is the (6x12) shape functions matrix.

In this case, we can also write the elastic displacement vector of the neutral axis as follows :

$$\begin{bmatrix} {}^{R_i}U_e(P) \\ {}^{R_i}\theta_e(P) \end{bmatrix} = \sum_{j=1}^{m_i} N_j^i(x)\, {}^{R_i}U_{ei}(j) \qquad (6)$$

with :

- $m_i = 12$, for $i = 1, \ldots, n$
- $N^i(x) = [N_1^i(x), N_2^i(x), \ldots, N_{12}^i]$
- $N_j^i(x) = [N_{j1}^i(x), \ldots, N_{6j}^i(x)]^T$

This new form allows us to deal with the finite elements model as well as with the modal expansion model.

Now, we adopt the general formulation :

$$\begin{bmatrix} {}^{R_i}U_e(P) \\ {}^{R_i}\theta_e(P) \end{bmatrix} = \sum_{j=1}^{m_i} \underline{\Phi}_j^i(x)\, q_{je}^i(t) = \underline{\Phi}^i(x)\, q_{ei}(t) \qquad (7)$$

where

- $q_{ei}(t)$ is the vector of modal or nodal coordinates,
- $\underline{\Phi}_i(x)$ is a matrix (6 x m_i).
- ${}^{R_i}U_e(P)^T = U^{iT} = [u_1^i(x), u_2^i(x), u_3^i(x)]$
- ${}^{R_i}\theta_e(P)^T = \underline{\theta}^{iT} = [\theta_1^i(x), \theta_2^i(x), \theta_3^i(x)]$
- $u_1^i(x)$, $u_2^i(x)$, $u_3^i(x)$, $\theta_1^i(x)$, $\theta_2^i(x)$ and $\theta_3^i(x)$ are the elastic displacements and rotations at abscissa x along the beam.

$$\begin{bmatrix} U^i \\ \underline{\theta}^i \end{bmatrix} = \underline{\Phi}^i(x)\, q_{ei}(t) = \begin{bmatrix} {}^t\underline{\Phi}^i(x) \\ {}^r\underline{\Phi}^i(x) \end{bmatrix} q_{ei}(t) \qquad (8)$$

where the left superscripts - " t " and " r " - are relative to the translation and rotation terms of the elastic displacements.

${}^t\underline{\Phi}^i(x)$ is a (3xm_i) matrix, m_i is the number of elastic degrees of freedom of link i :

$$ {}^{t}\underline{\Phi}^{1}(x) = \begin{bmatrix} \underline{\Phi}_1^1(x) \\ \underline{\Phi}_2^1(x) \\ \underline{\Phi}_3^1(x) \end{bmatrix} = \begin{bmatrix} \Phi_{11}^1(x) & & \Phi_{1mi}^1(x) \\ \Phi_{21}^1(x) & \dots & \Phi_{2mi}^1(x) \\ \Phi_{31}^1(x) & \dots & \Phi_{3mi}^1(x) \end{bmatrix} \quad (9) $$

The Euler-Bernoulli hypothesis allows us to write that $\theta_2^1(x) = -u_3^1(x)_{,x}$ and $\theta_3^1(x) = u_2^1(x)_{,x}$. Thus we can write ${}^{r}\underline{\Phi}^{i}(x)$ which is a (3xmi) matrix as :

$$ {}^{r}\underline{\Phi}^{1}(x) = \begin{bmatrix} \underline{\Phi}_4^i(x) \\ \underline{\Phi}_5^i(x) \\ \underline{\Phi}_6^i(x) \end{bmatrix} = \begin{bmatrix} \underline{\Phi}_4^1(x) \\ -\underline{\Phi}_3^1{}_{,x}(x) \\ \underline{\Phi}_2^i{}_{,x}(x) \end{bmatrix} = \begin{bmatrix} \Phi_{41}^1(x) & . . & \Phi_{4mi}^1(x) \\ -\Phi_{31}^1{}_{,x}(x) & . & -\Phi_{3mi}^1{}_{,x}(x) \\ \Phi_{21}^1{}_{,x}(x) & \dots & \Phi_{2mi}^i{}_{,x}(x) \end{bmatrix} \quad (10) $$

2.4. GEOMETRICAL AND KINEMATIC MODELS

The position in R_0 of any point M of C_i is given by the following expression :

$$ \begin{bmatrix} {}^{R0}OM \\ 1 \end{bmatrix} = {}^{R0}T_{R_i} \begin{bmatrix} {}^{R_i}L_iM \\ 1 \end{bmatrix} \quad (11) $$

with :

- $ {}^{R0}T_{R_i} = \prod_{j=0}^{i-1} {}^{R_j}T_{R_{j+1}} $
- ${}^{R_j}T_{R_{j+1}}$ is the homogeneous transformation matrix from R_j to R_{j+1}.

The kinematic model is derived from the geometrical model by differentiating equation (11).

$$ d\begin{bmatrix} {}^{R0}OM \\ 1 \end{bmatrix} = d{}^{R0}T_{R_i} \cdot \begin{bmatrix} {}^{R_i}L_iM \\ 1 \end{bmatrix} + {}^{R0}T_{R_i} \cdot d\begin{bmatrix} {}^{R_i}L_iM \\ 1 \end{bmatrix} \quad (12) $$

The calculation of $d{}^{R0}T_{R_i}$ is performed by using the differential homogeneous transformations. The general expression of $d{}^{R0}T_{R_i}$ is given in [2].

3. Energy Expression And Identification Model

3.1. LINEARIZED MODEL WITH RESPECT TO INERTIAL AND STIFFNESS PARAMETERS

In this section, we define the physical parameters which appear in the calculation of kinetic and potential energies of C_i. These parameters define the standard set of parameters of C_i. Equations (2), (3) and (7) lead to :

$$R_iL_iM = \begin{bmatrix} x + \sum_{j=1}^{m_i} \Phi_{j1}^i(x)\, q_{je}(t) + \sum_{j=1}^{m_i} (\Phi_{j5}^i(x)\, z - \Phi_{j6}^i(x)\, y)\, q_{je}(t) \\ y + \sum_{j=1}^{m_i} \Phi_{j2}^i(x)\, q_{je}(t) - \sum_{j=1}^{m_i} \Phi_{j4}^i(x)\, z\, q_{je}(t) \\ z + \sum_{j=1}^{m_i} \Phi_{j3}^i(x)\, q_{je}(t) - \sum_{j=1}^{m_i} \Phi_{j4}^i(x)\, y\, q_{je}(t) \end{bmatrix} \qquad (13)$$

The contribution of C_i in the total kinetic energy of the robot is defined as

$$Ec_i = \frac{1}{2} \int_{link\ i} (R_0\dot{O}M)^2\, dm \qquad (14)$$

where $R_0\dot{O}M = \frac{d}{dt}(R_0OM)$

The expanding of eq(14) using expressions (12) and (13) leads to a linear expression with respect to the following parameters of link i:

$$\int_{link\ i} \beta_1^2\, dm, \qquad \int_{link\ i} \beta_1\, \beta_2\, \Phi_{jk}^i(x)\, \Phi_{jl}^i(x)\, dm, \qquad \int_{link\ i} \beta_1\, \Phi_{jk}^i(x)\, dm$$

$$\int_{link\ i} \Phi_{jk}^i(x)\, \Phi_{jl}^i(x)\, dm, \qquad \int_{link\ i} \beta_1\, \beta_2 \Phi_{jk}^i(x)\, \Phi_{pl}^i(x)\, dm$$

with

- $(\beta_1, \beta_2) \in \{x, y, z\} \times \{x, y, z\}$
- $k = 1, \ldots, 6$; $l = 1, \ldots, 6$; $j = 1, \ldots, m_i$; $p = 1, \ldots, m_i$

In order to reduce the large number of parameters, we take into account a few hypotheses about the shape of the links. Thus, if the links are modeled as homogeneous and uniform beams of constant and symmetric cross section, the integral on link i becomes a product of an integral on the section and of an integral along the length of link i.

$$\int_{link\ i} \beta_1\, \beta_2 \Phi_{jk}^i(x)\, \Phi_{pl}^i(x)\, dm = \rho_i \int_{x_i'} \Phi_{jk}^i(x)\, \Phi_{pl}^i(x)\, dx \int_{S_i} \beta_1\, \beta_2\, dS \qquad (15)$$

If the length of the link is defined, the integrals of the functions Φ_{jk} or of other functionals of these functions (product, space derivative,....) are known.
The only unknown parameters are those which depend on the integral on the section : they

are defined by : $\int_{S_i} \beta_i \, \beta_2 \, dS$

Finally, the only unknown parameters are :

$$\rho_i S_i, \rho_i I_{xi}, \rho_i I_{yi} \text{ et } \rho_i I_{zi} \qquad (16\text{-a})$$

with

- ρ_i is the volumic mass of link i,
- S_i is the section of link i,
- I_{xi}, I_{yi} and I_{zi} are the quadratic momenta of the section.

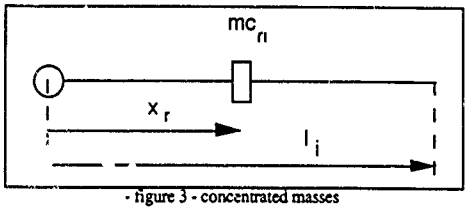
From the same hypotheses, it appears that the total potential energy - sum of potential and gravitational energies - depends linearly on the following parameters :

$$\rho_i S_i, E_i S_i, G_i I_{xi}, E_i I_{yi} \text{ and } E_i I_{zi} \qquad (16\text{-b})$$

where

- E_i is the Young's modulus of link i,
- G_i is the transverse modulus of link i.

We take into account p_i concentrated masses which are distributed along the neutral axis of the beam - figure 3- Therefore, we introduce p_i additional parameters which are the masses m_{cr} of these elements.



- figure 3 - concentrated masses

Finally, the standard set of parameters of a n-links robots is :

$$X = [\, X^{1T}, \ldots, X^{iT}, \ldots, X^{nT} \,]^T \qquad (17)$$

where

- X^i is a $((8 + p_i) \times 1)$ vector of link i parameters

$$X^i = \{\, \rho_i S_i, \rho_i I_{xi}, \rho_i I_{yi}, \rho_i I_{zi}, E_i S_i, G_i I_{xi}, E_i I_{yi}, E_i I_{zi}, m_{ci1}, \ldots, m_{cipi} \,]^T \qquad (18)$$

- m_{cij} is the j^{th} concentrated mass on the link C_i at the abscissa x_{cij}
- p_i is the total number of concentrated masses along link C_i.

3.2. ENERGY IDENTIFICATION MODEL

In order to identify the inertial and stiffness parameters of the robot, one can use either a dynamic model or an energy model. We suggest using the energy model in order to study

the minimality of the standard set of parameters because of the following advantages it uses scalar functions instead of the vector functions of the dynamic model and it avoids accelerations calculation [18].

The energy model is obtained from the total energy variation principle :

$$H(t_1) - H(t_2) = \Delta H(q, \dot{q}) = \int_{t_1}^{t_2} \Gamma^T \dot{q} \, dt \qquad (19)$$

where :
- $H = E + U$ is the total energy of the structure
- E is the kinetic energy
- U is the total potential energy. $U = Ed + Ep$
- Ed is the elastic potential energy.
- Ep is the gravity potential energy.
- $\dot{q}$ is the time derivative of q,
- $q = [q_1^T, \ldots, q_n^T]^T$ is the $(n + n_{e1})x1$ vector of the degrees of freedom of the structure
- $q_i = [\, q_{Ri}, q_{ei}^T]^T$,
- q_{Ri} is the joint variable of link i,
- q_{ei} is the $(m_i)x1$ vector of the elastic degrees of freedom (modal or nodal coordinates) of link i,
- $n_{e1} = \sum_{i=1}^{n} m_i$, is the number of elastic degrees of freedom,
- Γ is the vector of torques of the actuated joints

It is easy to verify that eq(19) can be rewritten linearly with respect to X .

$$\Delta H(q, \dot{q}) = \sum_{i=1}^{n} \Delta h^i . X^i = \int_{t_1}^{t_2} \Gamma^T \dot{q} \, dt \qquad (20)$$

where
- $\Delta h^i = h^i(t_1) - h^i(t_2)$
- $h^i(t) = [\, h_1^i(t), \ldots, h_{8+p1}^i(t) \,]$ is the coefficient of X^i in H.

4. Minimum Parameters

A classical identification method consists in a sampling of the linear model defined by eq(20) at different times ($t_i = 1, e$). We obtain an overdetermined system :

$$Y = W.X + b \qquad (21)$$

where

- Y is the (e x 1) measurements vector ; it corresponds to different values of $\int_{t_1}^{t_2} \Gamma^T \dot{q}$ dt at different instants t_1 and t_2.

- W is the (e x $\sum_{i=1}^{n} 8+p_i$) observation matrix which is related to the corresponding values of Δh^i_j,
- b is the(e x 1) errors vector.

The minimality of X is proved by calculating the rank of W : If W is a full rank matrix, X is minimal and the least square solution $\hat{X}$ is analytically given as the single solution of the normal equations :

$$[W^T W]\,\hat{X} = W^T X$$

In order to ensure the robustness of this identification process, we suggest to study the formal independence of the functions Δh^i_j. In the case of flexible structures, with the section 2 hypotheses, we offer conditions for regroupings which depend on the eigen functions (or shape functions).

4.1. FUNCTIONAL INDEPENDENCE

The expression (20) defined on a vector space E of functions Δh^i_j, - from $\mathbb{R}^{2(n+nei)}$ on $\mathbb{R}^n$. Thus we can formulate the search of a minimal set of parameters as the family of functions $\{\Delta h^i_j\}$ independence analysis .

In fact, if one can find a linear combination such as :

$$\Delta h^i_j = \sum_{r=1}^{n} \sum_{k=1}^{8+p_r} \alpha^r_k \Delta h^r_k \qquad \text{with } (r,k) \neq (i,j) \text{ and } \alpha^r_k \in \mathbb{R} \tag{22}$$

with α^r_k constant, the energy model is equivalent to the new model obtained by eliminating X^i_j and by replacing the X^r_k by $X^r_k + \alpha^r_k X^i_j$.

Let α_i be a ((8+pi)x1) vector :

$$\alpha_i = [\,\alpha^i_1, \ldots, \alpha^i_{8+pi}\,]^T$$

A necessary and sufficient condition for the family $\{\Delta h^i_j\}$ to be independent is :

$\forall \alpha^i \in \mathbb{R}^{8+pi}$ with $\forall\, i \in [1, n]$,

$$\sum_{i=1}^{n} \alpha^i \Delta h^i = 0 \;\Leftrightarrow\; \forall\, i \in [1, n],\;\; \alpha^i = 0 \tag{23}$$

where
- n is the number of links of the structure,
- $8+p_i$ is the dimension of the standard parameters vector of link i,
- p_i is the number of concentrated masses.

If the condition (23) is not verified, the family $\{\Delta h^i_j\}$ is linked and some linear combinations exist between the Δh^i_j. Some regroupings between parameters X^i_j are possible.

Remarks :

1) The functional independence must be verified for every couple $(q, \dot{q})$.
2) The condition (23) can be rewritten with h_j^i. Hence, we obtain :

$$\sum_{i=1}^{n} \alpha^i h^i + C = 0 \quad (\forall (\dot{q}, q)) \Leftrightarrow \alpha^i = 0 \,, \; i = 1, n \tag{24}$$

where C is a constant with respect to $(q, \dot{q})$, $C = \sum_{i=1}^{n} \alpha^i h^i$ for $t = t_1$

4.2. CONDITIONS OF REGROUPING. CASE OF A SINGLE LINK.

In what follows, we present the regrouping conditions of parameters of a link with parameters of the same link.

The functional independence of $\{\Delta h_j^i\}$ does not depend on the values of q and $\dot{q}$. Thus, it has to be verified for any set of couples $(q, \dot{q})$. It has to be verified particularly with $q_k = \dot{q}_k = 0$ for $k=1,.. \; i-1$ and $q_{Ri} = \dot{q}_{Ri} = 0$. If the motion of the link i is only parametrized with its elastic displacements, its kinetic and potential energies are given by the following expressions :

$$2\,Ec_i = \int_0^{l_i} \{\rho_i S_i \dot{U}^{i2}(x) + \rho_i I_{xi} \dot{\theta}_1^{i2}(x) + \rho_i I_{yi} \dot{\theta}_2^{i2}(x) + \rho_i I_{zi} \dot{\theta}_3^{i2}(x)\}\, dx \tag{25-a}$$

$$2\,Ed_i = \int_0^{l_i} \{E_i I_{yi} \theta_2^{i2},_x(x) + E_i I_{zi} \theta_3^{i2},_x(x) + E_i S_i u_1^{i2},_x(x) + G_i I_{xi} \theta_1^{i2},_x(x)\}\, dx \tag{25-b}$$

$$Ep_i = \int_0^{l_i} \rho_i S_i \; U^{iT}(x) \; g \, dx \tag{25-c}$$

where
- index 1 is the orientation of beam neutral axis, indexes 2 and 3 are the base vectors of the cross section.
- g is the gravity vector,

Similarly the kinetic energy of one concentrated mass m_{cr} at abscissa x_{cir} is given as :

$$Ecm_{cir} = \frac{1}{2} m_{cir} \dot{u}^{i2}(x_{cir}) \tag{26}$$

and its gravity potential energy as :

$$Epm_{cir} = m_{cir} u^{iT}(x_{cir}) \; g \tag{27}$$

From equations (18), (25), (26) and (27), we can extract the $8 + p_i$ coefficients h_j^i of the standard parameters :

$$h_{\rho_1 S_1} = \int_0^{l_1} (U^{1T}(x)\, g + \dot{U}^{1^2}(x))dx \quad (28\text{-a}) \qquad h_{\rho_1 I x_1} = \int_0^{l_1} \dot{\theta}_1^{1^2}(x)\, dx \quad (28\text{-b})$$

$$h_{\rho_1 I y_1} = \int_0^{l_1} \dot{\theta}_2^{1^2}(x)\, dx \quad (28\text{-c}) \qquad h_{\rho_1 I z_1} = \int_0^{l_1} \dot{\theta}_3^{1^2}(x)\, dx \quad (28\text{-d})$$

$$h_{EiIyi} = \int_0^{li} \theta_2^{1^2},_x(x)\, dx \quad (28\text{-e}) \qquad h_{E_1 I z_i} = \int_0^{l_1} \theta_3^{1^2},_x(x)\, dx \quad (28\text{-f})$$

$$h_{GiIxi} = \int_0^{l_1} \theta_1^{1^2},_x(x)\, dx) \quad (28\text{-g}) \qquad h_{EiS_1} = \int_0^{l_1} u_1^{1^2},_x(x)\, dx \quad (28\text{-h})$$

$$h_{mcir} = u^{1T}(x_{C1r})\, g + \frac{1}{2}\dot{u}^{i^2}(x_{C1r}) \quad r = 1 \ldots p_1 \quad (28\text{-i})$$

4 2.1. Expression Of The h_j^i as a Function Of $\Phi^i(x)$

Now, by using expressions (8), (9), (10) and (28), it is possible to write the $(8 + p_1)$ components h_j^i of the standard parameters with respect to the coefficients $\underline{\Phi}^i(x)$.

$$h_{\rho_1 Si} = \int_0^{li} (q_{1e}{}^T\, {}^t\underline{\Phi}^{1T}(x)\, g + \dot{q}_{1e}{}^T\, {}^t\underline{\Phi}^{1T}(x)\, {}^t\underline{\Phi}^{i}(x)\, \dot{q}_{ie})\, dx$$

$$= q_{1e}{}^T\, G_{\rho_1 S_1} + \dot{q}_{ie}{}^T\, A_{\rho i S_1}\, \dot{q}_{ie} \quad (29\text{-a})$$

$$h_{\rho_i I x_1} = \int_0^{l_1} \dot{q}_{1e}{}^T\, \underline{\Phi}_4^{1T}(x)\, \underline{\Phi}_4^{1}(x)\, \dot{q}_{ie}\, dx = \dot{q}_{ie}{}^T\, A_{\rho_1 I x_1}\, \dot{q}_{ie} \quad (29\text{-b})$$

$$h_{\rho_1 I y_i} = \int_0^{li} \dot{q}_{1e}{}^T\, \underline{\Phi}_{3,x}^{1T}(x)\, \underline{\Phi}_{3,x}^{1}(x)\, \dot{q}_{ie}\, dx = \dot{q}_{ie}{}^T\, A_{\rho_1 I y_1}\, \dot{q}_{ie} \quad (29\text{-c})$$

$$h_{\rho_i I z_i} = \int_0^{li} \dot{q}_{1e}{}^T\, \underline{\Phi}_{2,x}^{iT}(x)\, \underline{\Phi}_{2,x}^{i}(x)\, \dot{q}_{ie}\, dx = \dot{q}_{ie}{}^T\, A_{\rho_1 I z_i}\, \dot{q}_{ie} \quad (29\text{-d})$$

$$h_{E_1 I y_i} = \int_0^{li} q_{ie}{}^T\, \underline{\Phi}_{3,xx}^{iT}(x)\, \underline{\Phi}_{3,xx}^{i}(x)\, q_{ie}\, dx = q_{ie}{}^T\, K_{E i I y_1}\, q_{ie} \quad (29\text{-e})$$

$$h_{E_1 I z_i} = \int_0^{li} q_{ie}{}^T\, \underline{\Phi}_{2,xx}^{1T}(x)\, \underline{\Phi}_{2,xx}^{1}(x)\, q_{ie}\, dx = q_{1e}{}^T\, K_{E_1 I z_1}\, q_{1e} \quad (29\text{-f})$$

$$h_{GiIx1} = \int_0^{l_i} q_{ie}^T \, \underline{\Phi}_{4,x}^{i\,T}(x) \, \underline{\Phi}_{4,x}^{i}(x) \, q_{ie} \, dx = q_{ie}^T K_{GiIx1} \, q_{ie} \qquad (29\text{-g})$$

$$h_{EiSi} = \int_0^{l_i} q_{ie}^T \, \underline{\Phi}_{1,x}^{i\,T}(x) \, \underline{\Phi}_{1,x}^{i}(x) \, q_{ie} \, dx = q_{ie}^T K_{EiSi} \, q_{ie} \qquad (29\text{-h})$$

$$h_{mcir} = \frac{1}{2} \dot{q}_{ie}^T \left({}^t\underline{\Phi}^{i\,T}(x_{cir}) \, {}^t\underline{\Phi}^{i}(x_{cir}) \right) \dot{q}_{ie} + q_{ie}^T \, {}^t\underline{\Phi}^{i\,T}(x_{cir}) \, g$$

$$= q_{ie}^T G_{mcir} + \dot{q}_{ie}^T A_{mcir} \, \dot{q}_{ie} \qquad r = 1,.. \, p_i \qquad (29\text{-i})$$

with :

$$A_{\rho iSi} = \int_0^{l_i} \left[\underline{\Phi}_1^{i\,T}(x) \, \underline{\Phi}_1^{i}(x) + \underline{\Phi}_2^{i\,T}(x) \, \underline{\Phi}_2^{i}(x) + \underline{\Phi}_3^{i\,T}(x) \, \underline{\Phi}_3^{i}(x) \right] dx$$

$$A_{\rho iIx1} = \int_0^{l_i} \underline{\Phi}_4^{i\,T}(x) \, \underline{\Phi}_4^{i}(x) \, dx ; \qquad A_{\rho iIy1} = \int_0^{l_i} \underline{\Phi}_{3,x}^{i\,T}(x) \, \underline{\Phi}_{3,x}^{i}(x) \, dx$$

$$A_{\rho iIz1} = \int_0^{l_i} \underline{\Phi}_{2,x}^{i\,T}(x) \, \underline{\Phi}_{2,x}^{i}(x) \, dx$$

$$A_{mcir} = \underline{\Phi}_1^{i\,T}(x_{cir}) \, \underline{\Phi}_1^{i}(x_{cir}) + \underline{\Phi}_2^{i\,T}(x_{cir}) \, \underline{\Phi}_2^{i}(x_{cir}) + \underline{\Phi}_3^{i\,T}(x_{cir}) \, \underline{\Phi}_3^{i}(x_{cir}) \qquad r = 1,.. \, p_i$$

$$K_{EiSi} = \int_0^{l_i} \underline{\Phi}_{1,x}^{i\,T}(x) \, \underline{\Phi}_{1,x}^{i}(x) \, dx ; \qquad K_{EiIy1} = \int_0^{l_i} \underline{\Phi}_{3,xx}^{i\,T}(x) \, \underline{\Phi}_{3,xx}^{i}(x) \, dx$$

$$K_{EiIz1} = \int_0^{l_i} \underline{\Phi}_{2,xx}^{i\,T}(x) \, \underline{\Phi}_{2,xx}^{i}(x) \, dx ; \qquad K_{GiIxi} = \int_0^{l_i} \underline{\Phi}_{4,x}^{i\,T}(x) \, \underline{\Phi}_{4,x}^{i}(x) \, dx$$

$$G_{\rho iSi} = \int_0^{l_i} {}^t\underline{\Phi}^{i}(x_{cir})^T \, g \, dx ; \qquad G_{mcir} = {}^t\underline{\Phi}^{i}(x_{cir})^T \, g \qquad r = 1,.. \, p_i$$

The dimensions of the matrices A_a, K_k and G_g are respectively $(m_i \times m_i)$, $(m_i \times m_i)$ and $(m_i \times 1)$.

We note that the link i total energy H^i can be written as :

$$H_i = \dot{q}_{ie}^T A^i \dot{q}_{ie} + q_{ie}^T K^i \, q_{ie} + q_{ie}^T G^i$$

where

- A^i is a constant matrix :

$$A^i = A_{\rho iSi} \, \rho_i S_i + A_{\rho iIx1} \, \rho_i I_{xi} + A_{\rho iIy1} \, \rho_i I_{yi} + A_{\rho iIz1} \, \rho_i I_{zi} + \sum_{r=1}^{p_i} A_{mcir} \, m_{cir}$$

- K^i is a constant matrix :

$$K^i = K_{EiSi} \, E_i S_i + K_{EiIy1} \, E_i I_{yi} + K_{EiIz1} \, E_i I_{zi} + K_{GiIx1} \, G_i I_{xi}$$

- G^i is a constant matrix :

$$G^i = G_{\rho i S i}\, \rho_i S_i + \sum_{r=1}^{p_i} G_{mcir}\, m_{cir}$$

4 2 2. Regrouping Conditions

If a linear combination exists between the functions of energy h^i of the link i, it can be written as follows :

$$\underline{\alpha}^i\, h^i + C = 0$$

Rewritten with respect to the matrices A_a, K_k and G_g, this expression leads to a linear combination between these matrices :

$$\dot{q}_{ie}^T \{\alpha_{\rho i S i}\, A_{\rho i S i} + \alpha_{\rho i I x i}\, A_{\rho i I x i} + \alpha_{\rho i I y i}\, A_{\rho i I y i} + \alpha_{\rho i I z i}\, A_{\rho i I z i} + \sum_{r=1}^{p_i} \alpha_{mcir}\, A_{mcir}\}\, \dot{q}_{ie}$$

$$+ q_{ie}^T \{\alpha_{Ei S i}\, K_{Ei S i} + \alpha_{Ei I y i}\, K_{Ei I y i} + \alpha_{Ei I z i}\, K_{Ei I z i} + \alpha_{Gi I x i}\, K_{Gi I x i}\}\, q_{ie}$$

$$+ q_{ie}^T \{\alpha_{\rho i S i}\, G_{\rho i S i} + \sum_{r=1}^{pi} \alpha_{mcir}\, G_{mcir}\} + C = 0 \qquad (30)$$

However, by definition, the modal or nodal coordinates -q_{ie}- and their time derivatives -$\dot{q}_{ie}$- are independent. Hence, eq(30) leads to the following linear system of equations :

$$\alpha_{\rho i S i}\, A_{\rho i S i} + \alpha_{\rho i I x i}\, A_{\rho i I x i} + \alpha_{\rho i I y i}\, A_{\rho i I y i} + \alpha_{\rho i I z i}\, A_{\rho i I z i} + \sum_{r=1}^{pi} \alpha_{mcir}\, A_{mcir} = 0$$

$$\alpha_{\rho i S i}\, G_{\rho i S i} + \sum_{r=1}^{p_i} \alpha_{mcir}\, G_{mcir} = 0$$

$$\alpha_{Ei S i}\, K_{Ei S i} + \alpha_{Ei I y i}\, K_{Ei I y i} + \alpha_{Ei I z i}\, K_{Ei I z i} + \alpha_{Gi I x i}\, K_{Gi I x i} = 0 \qquad (31)$$

To each symetric matrix A_a and K_k can be associated a vector A_a^* and K_k^*, obtained by ordering one by one and in a predefined order the components of A_a in A_a^* and the components of K_k in K_k^*.

The dimension of the A_a^* and K_k^* matrices is $\left(\frac{m_i (m_i+1)}{2} \times 1\right)$.

Finally, we obtain the following system of equations :

$$D^i\, \underline{\alpha}^i = \begin{bmatrix} A^i_{ine} & 0 \\ 0 & K^i_{raid} \end{bmatrix} \begin{bmatrix} \alpha^i_{ine} \\ \alpha^i_{raid} \end{bmatrix} = 0 \qquad (32)$$

with :

$$A_{ine}^{1} = \begin{bmatrix} 0 & 0 & 0 & G_{\rho 1 Si} & G_{mc11} & G_{mc12} & . & G_{mc1p1} \\ A'_{\rho 1 Ix1} & A^{*}_{\rho 1 Iy1} & A^{*}_{\rho 1 Iz1} & A^{*}_{\rho iSi} & A^{*}_{mc11} & A^{*}_{mc12} & & A^{*}_{mc1p1} \end{bmatrix}$$

$$K_{raid}^{1} = \begin{bmatrix} K^{*}_{E1Si} & K^{*}_{E1Iy1} & K^{*}_{E1Iz1} & K^{*}_{G1Ix1} \end{bmatrix}$$

$$\alpha_{ine}^{1} = \begin{bmatrix} \alpha_{m}^{1} \\ \alpha_{mc}^{1} \end{bmatrix}$$

$$\alpha_{m}^{1} = [\, \alpha_{\rho 1 Ix1} ; \alpha_{\rho 1 Iy1} ; \alpha_{\rho 1 Iz1}; \alpha_{\rho 1 Si} \,]^{T}$$

$$\alpha_{mc}^{1} = [\alpha_{mc11} ; \alpha_{mc12} , \ldots ; \alpha_{mc1p1}]^{T}$$

$$\alpha_{raid}^{1} = [\alpha_{E1Si} ; \alpha_{E1Iy1} , \alpha_{E1Iz1} ; \alpha_{G1Ixi} \,]^{T}$$

From eq(32), we can formulate two theorems :

Theorem 1 :
If the rank - c- of the matrix D^i is equal to (8+pi), there is no regrouping. Therefore, there are (8+pi) minimal parameters.

Demonstration :
With this condition, the matrix of the system described by eq(32) is a full rank matrix and the single solution of this system is the null vector. There is no linear combination between link's parameters and no regroupings.

Theorem 2 :
If the rank - c- of the matrix D^i is lower than (8+p1), there are some linear combinations between the functions h_i^j which lead to $(8 + p_1 - c)$ regroupings

Demonstration :
c is lower than $(8 + p_1)$: c is the maximun number of lines or columns of D^i which are independent. The dimension of the vector space E is c. We can build a base of this space with the c independent columns.

If there are some regroupings, the minimum set of parameters becomes XR_i. Its dimension is c. XR_i is obtained by selecting the principal parameters which fit with the minor of maximal rank extracted from D^i.

Remarks :
- As one can notice that matrix D^i is a bloc diagonal matrix - see eq(32) -, the condition of regrouping which depends on the rank of D^i, leads to the two following conditions :

$\text{rank}(A_{ine}^{i}) = 4 + p_1$ et $\text{rank}(K_{raid}^{i}) = 4$

- m_i is the number of elastic degrees of freedom

* $\text{rank}(A_{ine}^{i}) = 4 + p_i \Longrightarrow \dfrac{m_i\,(m_i+3)}{2} > 4 + p_1$

* $\text{rank}(K^1_{raid}) \geq 4 \Longrightarrow \frac{m_i (m_i+1)}{2} \geq 4 \Longrightarrow m_i \geq 3$

If $m_i = 3$ and $p_i > 5$, we are sure that a regrouping exists :

$$\frac{m_i (m_i+3)}{2} = 9 \text{ and } 4 + p_i > 9$$

If there are some regroupings, we only keep functions Δh^1_k which are independent and not equal to zero. If $\Delta h^1_k = 0$, H does not depend on X^1_k. Now we suppose that the modal (or shape) functions are chosen in order to prevent the possibility for any Δh^1_k ($i=1,..,n$; $k=i,...,8+p_i$) to be equal to zero.

4.3 CONDITIONS OF REGROUPING. CASE OF N LINKS.

We suppose that all the possible regroupings described before have been done. They lead to the independence of the energy functions for each link.
In this section we demonstrate that there is no regrouping of a parameter of one link with some parameters of another link. The demonstration is based on a recurrence from link n to link 1. We have :

$$\sum_{i=1}^{n} \underline{\alpha}^i \Delta h^i = 0 \Leftrightarrow \alpha^n \Delta h^n + \sum_{i=1}^{n-1} \alpha^i \Delta h^i = 0 \qquad (33)$$

Expression (33) must be verified for any value of $(q, \dot{q})$. One can choose two instants t_1 and t_2 such as :

$$\forall\, i = 1, n-1 \qquad \begin{cases} \dot{q}_{Ri}(t_1) = \dot{q}_{Ri}(t_2) \text{ and } q_{Ri}(t_1) = q_{Ri}(t_2) \\ \dot{q}_{ei}(t_1) = \dot{q}_{ei}(t_2) \text{ and } q_{ei}(t_1) = q_{ei}(t_2) \end{cases} \qquad (34)$$

With this couple $(q, \dot{q})$, we can notice that

$$\forall\, i = 1, n-1 \quad \Delta h^i = 0$$

because the components of h^i depend only on the components of the vectors $\dot{q}_j$ et q_j ($j = 1, i$). In fact, the total energy H^i of the link is a function of the standard parameters vector - X^i - and of the components of $\dot{q}_j$ and q_j ($j = 1, i$).
Thus, expression (33) and conditions (34) lead to :

$$\underline{\alpha}^n \Delta h^n = 0 \qquad (35)$$

First it appears that there is no possible regrouping of parameters of the link n with the antecedent links parameters.
Moreover, we have $\underline{\alpha}^n = 0$ thanks to the results of section 4.2.
So, there is no linear combination between the parameters of the link n and the parameters of the other links. The only possible regroupings of parameters of link n are regroupings of parameters of this link with some parameters of the same link.

In the same way, we demonstrate by recurrence (k = n-1 to k = 1) that for the whole structure the only possible regroupings are regroupings of parameters of one link between them.

5. Conclusion

We have just shown that it is possible to study the minimality of the dynamic parameters of a flexible robot.
The approach we have chosen is general and symbolic. This approach can be applied to any type of robot. It is symbolic because it is not necessary to simulate a random trajectory in order to find a sequence of points. Our approach is based on an analysis of the independence of the energy functions.
Finally we have presented a method which leads to a necessary and sufficient condition (theorem 1) which ensures the minimality of the set of standard parameters of a flexible robot. These conditions depend on eigen functions if the model comes out from an assumed modes method and on shape functions if the model comes out from finite elements method.
Especially the dimension of the minimal set of parameters is linked to the number of modes used to describe links' flexibilites.
Lastly, in the case of robots whose links are all flexible, we prove that the only possible regroupings of parameters are provided with parameters belonging to the same link.
Further developments of this research deal with the identification process. Results have been obtained previously on a two flexible links planar robot. The identification is in progress in the case of a three links anthropomorphic robot.

6. References

[1] Geradin, M., Cardona, A., "Kinamatics and dynamics of rigid and flexible mechanics using finite elements and quaternion algebra". Computational Mechanics, Vol. 4, pp 115-135, 1989.
[2] Chedmail, P., "Synthèse de robots et de sites robotisés - Modélisation de robots souples". Thèse de Doctorat d'Etat, Université de Nantes, E.N.S.M., janvier 1990.
[3] Singer, N. C., Seering, W. P., "Design and comparison of command shaping methods for controlling residual vibration". Proc IEEE Conf. on Robotics and Automation, Scottsdale, Phoenix, May 1989, pp 888-893.
[4] Meldrum, D. R., Balas, M. J., "Application of model reference adaptative control of a flexible remote manipulator arm". Proc. of American Control Conference, Summer, 1986, pp 825-832.
[5] Siciliano, B. Book, W.J., "A singular perturbation approach to control of lightweight flexible manipulators". International Journal of Robotics Research, Vol.7, No. 4, August 1988, pp. 79-90.
[6] Chaloub, N. G., Ulsoy, A. G., "Control of a flexible robot arm : Experimental and theoretical results". Transactions of the ASME Journal of Dynamic systems Measurement and Control, Décembre 87, Vol 109, pp 299-310.
[7] Chedmail, P., Aoustin, Y., Chevallereau, C., "Modelling and control of flexible robots". International Journal of Numerical Methods in Engineering, John Wiley and Sons, décembre 1991, Vol. 32, pp 1595-1619

[8] Aoustin, Y., Chedmail, P., Glumineau, A., "Some simulation results on the robustness of a flexible arm nonlinear control law". International Journal of Modelling and Simulation, vol. 11, issue 4, 1991.
[9] Parks, T. R., Pak , H. A , " Effect of payload on the dynamics of a flexible manipulator - Modeling for control". Journal of dynamics systems, measurement, and control., 1991, Vol. 113, pp 409-418.
[10] Gautier, M., Khalil, W., "Direct calculation of minimum set of inertial parameters of serial robots". IEEE Trans. on Robotics and Automation, Vol 6, N° 3, pp 368-373, 1990.
[11] Gautier, M., "Numerical calculation of the base inertial parameters". Proc. IEEE Trans. on Robotics and Automation, Cincinnan, pp 1020-1025, 1990.
[12] Bennis, F., Khalil, W., Gautier, M. "Calculation of the base inertial parameters of closed-loops robots". Proc. IEEE Trans. on Robotics and Automation, Nice, pp 370-375, 1992.
[13] Sheu S. Y. et Walker M. W., "Basis sets for manipulator inertial parameters". Proc IEEE Conf. on Robotics and Automation, Scottsdale, Phoenix. May 1989, pp 1517-1522.
[14] Mayeda, H., Yoshida, K., Ohashi, K., " Base parameters of dynamic models for manipulators with rotational and translational joints". Proc. IEEE conference on Robotics and Automation, Scottsdale, 1989, pp 1523-1528.
[15] Tzes, A. P., Yurkovich, S., "Application and comparison of on-line identification methods for flexible manipulator control". The International Journal of Robotics Research, Vol. 10, N° 5, October 1991, pp 515-527.
[16] Rovner, D. M., Cannon, R. H., "Experiments toward on-line identification and control of a very flexible one-link manipultor". The International Journal of Robotics Research, Vol. 6, N° 4, Winter 1987, pp 3-19.
[17] Cetinkunt, S., Wu, S., " Tip position control of a flexible one arm robot with predictive adaptative output feedback implemented with lattice filter parameter identifier". Proc. IEEE Trans. on Robotics and Automation, Cincinnati, pp 1620-1625, 1990
[18] Gautier, M., Pham, C. M., Chedmail, P., "Energy based determination of identifiable parameters of flexible link robots". IFAC/SYROCO'91, Vienne (Autriche), pp 399-404, 1991.
[19] Chedmail P., Gautier M. et Pham C. M., "Determination of base parameters of flexible links manipulators". Proc. of IMACS-Symposium on Modeling and Control of Technological Systems, Lille, 1991, pp 524-529.

SAFETY AND SURVIVABILITY ANALYSIS OF WALL-CLIMBING ROBOT

Behnam Bahr, Ph.D.
Guanghua Li, Graduate Student
Wichita State University
Mechanical Engineering Department
Wichita, KS 67260-0035, USA

ABSTRACT: In this paper, an explanation of the need for the climbing robot is given first. Second, a safety analysis is performed to illustrate how the robot is guarded against falling. This safety analysis is used to determine the design forces required on the suction cups that hold the robot in place. Finally, a survivability analysis was done to show that the robot can withstand a drop from a height of 1 3 meters if it were to fall If the robot does fall, it would be desirable to minimize the amount of damage on the equipment mounted on the robot. The method presented in this paper proposes to swing the arms of the robot into the direction of the impact, using them to absorb the majority of the impact energy. For analysis of the fall protection system, a multibody model was developed to simulate its dynamics during a fall.

1. Introduction

Robot manipulators are well suited for structured environments and are currently being used for most modern manufacturing operations. Researchers are continually looking to expand the application of robots in other areas. Ordinary robotic manipulators used in normal industrial applications are not suitable for unstructured environments such as nuclear power plants, chemical plants, tall buildings, bridges, ships, and aircraft. Robots needed for unstructured environments must be mobile, intelligent, and be able to perform tasks that are dangerous, difficult, and tedious for human operators However, these mobile robots have some limitations To extend the capabilities of these robots, a relatively new and exciting idea for climbing robot is being developed. This climbing robot can move on all surfaces -- walls, ceilings, and floor. Since climbing robots have the capability to move around freely, they have the most potential for use in unstructured environments, performing tasks such as cleaning of a tall building or repairing a ship.

Mobile robots can replace humans or reduce the exposure of humans in hazardous environments. They can also relieve workers from repetitive work. Research on mobile robots and legged locomotion has been pursued for many years, and there have been many excellent contributions made in these areas. Hemami [1] considered the stability

of a mobile robot on rough terrain. Waldron [2] studied the kinematic analysis of the legged robots and developed an algorithm of motion planning for a six legged robot. Messuri et al. [3] discussed two types of walking mechanisms, one with control of multi-joint legs and the other with decoupled degrees of freedom. He showed that multi-joint legs would be able to negotiate rough terrain better than the machines that walk with the fixed pattern. Song [4] described an efficient analytical approach for gait study and an application for sequencing and placing the legs. McGhee [5], Isik [6] and Crowley [7] investigated computer control of legged robots on various terrains. However, the mobile robots used in the above studies do not have the reach capability that one may require in an unstructured environment. Due to the deficiencies of these mobile robots, there is a need for another class of robotic systems such as a climbing robot. Climbing robots can be designed to move or climb on almost any surface that is horizontal, inclined, vertical, and even upside-down on a ceiling.

Robots that can attach to and climb on many structure surfaces are just beginning to emerge. Research on mobile climbing robots for inspection and maintenance is being pursued in Japan [8, 9, 10, 11], France [12], Russia [13], and recently in the U.S. [14, 15, 16]. A wall-climbing robot built by Hirose [8] uses magnets which allow the robot to move on the steel walls of steam boilers in nuclear power plants or ships. Akir, et al. [9, 10, 11] designed and demonstrated several climbing robot, one of which could climb a vertical wall of a building using a turbo type fan system to create adhesion and driving forces.

The climbing robot presented in this paper is a succession of the ROSTAM (Robotics System for Total Aircraft Maintenance), which was devised at Wichita State University [14, 15, 16]. Its predecessors include ROSTAM I, a four-legged robot with suction cups on each of the legs as shown in Figure 1. This robot can move in two directions perpendicular to each other or diagonally in a series of steps in either perpendicular directions. This robot can also travel on a slightly curved surface. ROSTAM II, a two-legged robot shown in Figure 2, can also travel on a curved surface, but it can only move in one direction. The advantage of ROSTAM II is that its lighter weight. The next generation ROSTAM III, shown in Figure 3, has a central rotation system for changing the moving direction to any desired angle. All three of these robotic systems use suction cups for adhesion. The suction cups were adopted, since they can adhere to many more material types than the magnetic method.

The safety analysis is performed so that the robot does not fall. However, if it were to fall, the simulation results are presented through animation and graphical representations. There are many animation programs in today's software market, such as Autodesk animator pro [17], which uses a movie method, but this kind of animator is not suitable for mechanical system animation. For mechanical system animation, a physics-based animation needed to be developed. Thus an animation/simulation system based program was developed [18, 19]. Using AutoCAD, the geometry of a complex mechanical system can be easily modeled and modified using the graphics database interface. A new graphics database can be produced for each animation frame by processing geometric information from a graphical database and simulation results.

Figure 1 ROSTAM I, four legged robot.

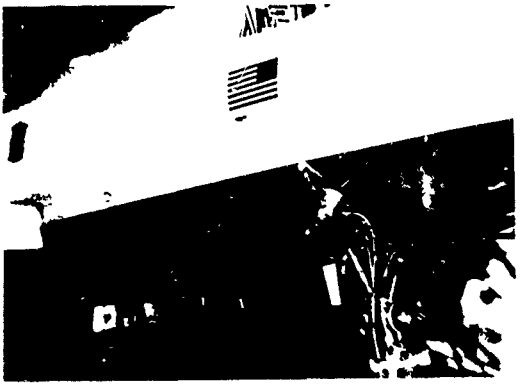


Figure 2 ROSTAM II, the two legged robot

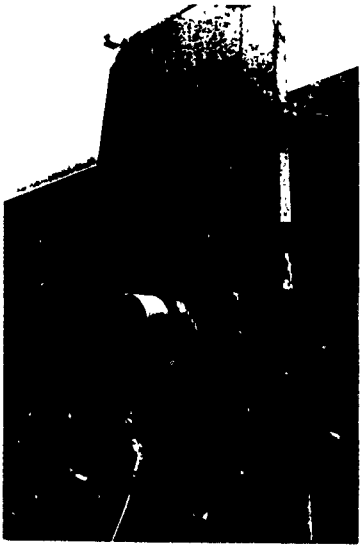


Figure 3 ROSTAM III, two legged robot with rotational center

2. Robot's movement commands and specifications

As shown in Figure 3, there are two suction cups allocated on each side of the robot's two legs, and there are four suction cups in the center portion of the robot The suction vacuum generators are connected to each suction cups for safety reasons Two air cylinders comprise the legs which are used for creating motion of the robot The rotation of the robot is with respect to its base in the moving direction This motion is generated by a DC motor through a pair of worm gear systems The central portion of the robot sticks to the surface while the two legs rotate about it to a desired direction The suction cups on two legs can work either synchronously (both legs are sticking or moving), or asynchronously (one leg is sticking to hold the robot and the other leg is making a forward movement, and then attaching its suction cups on the surface to hold

$$f' = f(q_A, q_B, y') \tag{5}$$

Where q_A, q_B are the reaction force density at extreme point A and B respectively

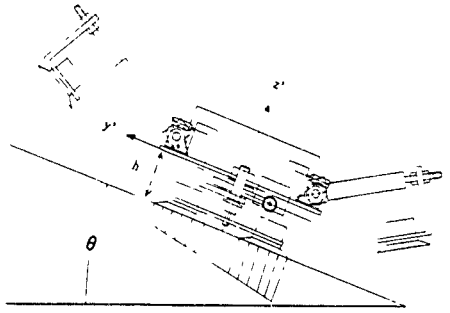


Figure 4 Safety analysis model of the robot.

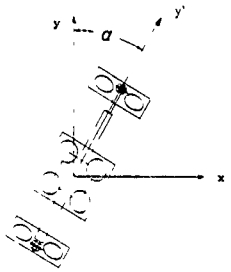


Figure 5 Safety analysis of the robot moving in the α direction

3. Safety Analysis of the Robot

The most important consideration in designing the climbing robot is to ensure that it is secured firmly on the surface while climbing. There are two dangerous circumstances that could occur when the robot is climbing on a surface. One is slipping and the other is falling. In general, the robot can be modeled for the safety analysis while climbing an inclined surface as shown in Figures 4 and 5. In Figures 4 and 5, Θ represents the inclined angle, and α stands for the moving direction. When Θ equals 90°, it means the robot is moving on the vertical plane, and when Θ equals 180°, it means the robot is on the ceiling.

The tipping moment which would cause the robot to fall can be expressed in xyz coordinate system as:

$$M_x = m\,g\,h\,\sin\theta + M_d\cos\alpha \qquad (1)$$

$$M_y = M_d \sin\alpha \qquad (2)$$

Where

m is mass of the robot;
M_d is the dynamic moment caused by the impact of the cylinder;
g gravitational acceleration,
h height of the mass center of the robot.

The equivalent total tipping moment and its directional angle can be obtained as.

$$M = \sqrt{M_x^2 + M_y^2} \qquad (3)$$

$$\beta = \tan^{-1}(M_y / M_x) \qquad (4)$$

Using the above equivalent total tipping moment, the equilibrium equation will be created so that, except along the line with β degree, there is no active moment.

As it is observed from Figure 4, the highest point on the suction cups (the point with maximum y′ coordinate in x′y′z′ system) has the most and earliest tendency to leave the surface, and the lowest point (the point with the least y′ coordinate in x′y′z′ system) has the least. Therefore, it is reasonable to assume that the reaction force acting on the suction cups has linear distribution along y′ axis shown in Figure 4. Based on this assumption, the reaction force density (f′) on every point of the suction cups can be expressed as a linear function of its y′ coordinate in x′y′z′ coordinate system and the amount of the reaction force density at extreme points A, and B or

the robot) The detailed working procedure for synchronous operation is explained as following:

(1) Initialization consists of all the suction cups sticking, one leg is in a retract position, and the other in extended position, as shown in Figure 3.
(2) If more steps are required continue, otherwise stop the movement.
(3) Check the vacuum on the central suction cups If vacuum is not good stop, otherwise continue
(4) Release the suction on the legs.
(5) Move the legs away from the surface for a predetermined amount of time
(6) If desired, rotate the legs in a new direction
(7) Extend one leg and retract the other leg
(8) Start the vacuum on the legs
(9) Move the legs so that they touch the surface
(10) Check the amount of vacuum on the legs suction cups, if vacuum is not good stop, otherwise continue.
(11) Release the vacuum on the central suction cups.
(12) Extend one leg and retract the other one At this time, the central portion of the robot will move one step.
(13) Start the vacuum on the central suction cups
(14) Go to step 2.

The voltage used for DC motors and solenoid valves is 24 volts The diameter of each suction cup is 11 4 cm. The suction cups can create 34 5 kg force under 18 Hg vacuum according to the component catalog [20] More specifications of the robot are given in the table 1.

Table 1 Specifications of the robot.

Robot parts	Weight (kg)	Length (cm)	Height from surface (cm)
Central body	8.5	20	14
Left leg	1.0	9	8
Right leg	1.0	6	8

The reaction force acting on a suction cup "i" can therefore be written as:

$$f_i = \int_{y_{c_i}'-\frac{d}{2}}^{y_{c_i}'+\frac{d}{2}} f' \, dy' = \int_{y_{c_i}'-\frac{d}{2}}^{y_{c_i}'+\frac{d}{2}} f(q_A, q_B, y') \, dy' \tag{6}$$

The moment of each suction cup contributed by the reaction force can be expressed as

$$\begin{aligned} M_{ix} &= f_i \, x_{ci}' \\ M_{iy} &= f_i \, y_{ci}' \end{aligned} \tag{7}$$

Where x'_{ci}, and y'_{ci} are the x and y coordinates of the center of the suction cup "i" in x'y'z' coordinate system; and d is the diameter of the suction cup.

The force and moment equilibrium equations are derived as.

$$\sum_{i=1}^{n} F \, y_{c_i}' - \sum_{i=1}^{n} M_{ix} - M \cos(\alpha+\beta) = 0 \tag{8}$$

$$\sum_{i=1}^{n} F \, x_{c_i}' - \sum_{i=1}^{n} M_{iy} - M \sin(\alpha+\beta) = 0 \tag{9}$$

$$m \, g \cos\vartheta + n \, F + \sum_{i=1}^{n} f_i = 0 \tag{10}$$

Where

F is the suction force produced by a single suction cup;
y'_{ci} is the y coordinate of the center of suction cup "i" in x'y'z',
f_i is the perpendicular reaction force acting on suction cup "i";
M is the moment causing the robot to fall, which was expressed in equation (3),
n is the number of suction cups holding the robot.

When robot is in safe condition, the perpendicular reaction force which is acting on every point of the edge of the suction cups should be greater than zero. The critical case occurs when these reaction forces become zero; for example when q_A becomes equal to zero. This means there is no contact between the suction cup edge and the surface, and the robot will fall. The result of the safety analysis is shown in the Figure 6 The most dangerous condition for this robot is when it is climbing and inclined surface with the angle of about 150 degrees, and moving in a 45 degrees direction as can be seen in the Figure 6.

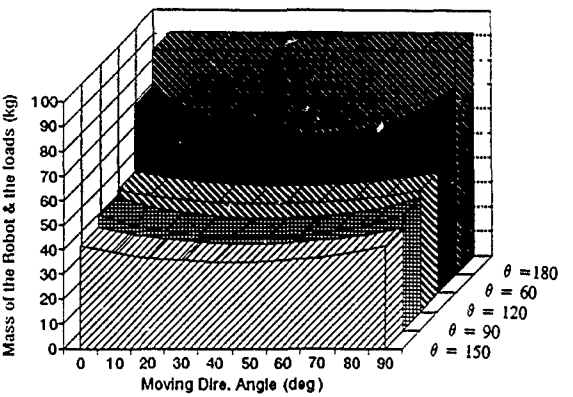


Figure 6 Safety analysis results

4. Survivability Analysis of the robot

The robot under development is designed to operate autonomously performing tasks in areas difficult to access with a human operator. One such task is performing external inspections on aircraft skin joints. On an aircraft like the Boeing 747, it would be possible for the robot to fall over 3 meters unobstructed (not including the tail section). A 3D simulation/animation program has been developed so that the robot response as a result of a fall is observed. The goal was to create a geometry model which uses dynamic performance of the robot to control its motion during the fall. From the dynamic analysis results, graphical animations of the robot were produced. In order to obtain realistic representation of the robot, a shading technique was employed. The developed 3D animation algorithm is geometry independent; hence it can be used to animate variety of different mechanical systems without rewriting or modifying the program.

The robot under consideration is sometimes an open-loop and sometimes a closed loop system. Hence the differential equations of the motion of the robot were formulated using Cartesian coordinates. The use of Cartesian coordinates and the resulting formulation of the equation of motion in a Newtonian form allow the modelling of both

open and closed chain systems in a simple manner Furthermore, it enhances the simplicity of the numerical formulation or generation of the governing equations, and the numerical development of the solutions. A general body "i" of a multibody system under consideration, such as the robot, is shown in Figure 7 A set of local body-fixed coordinate axes $\xi\eta\zeta$ with origin at the element mass center C, is attached to the body To describe the configuration of the body in a non-moving reference frame xyz, it is sufficient to specify the spatial location of point C, and the angular orientation of the local axes A vector of Cartesian coordinates for the body, q_i, contains $r_i = [x, y, z]_i^T$ and $p_i = [e_0, e^T]^T = [e_0, e_1, e_2, e_3]^T$ The four parameters e_0, e_1, e_2 and e_3 form a set of normalized quaternions, known as Euler parameters describing the rotational configurations [19]. The parameters are described below as:

$$e_0 = \cos(\frac{\phi}{2}) \tag{11}$$

$$\vec{e} = \vec{u} \sin(\frac{\phi}{2}) \tag{12}$$

$$e_0^2+e_1^2+e_2^2+e_3^2 = 1 \tag{13}$$

where $\vec{u}$ describe the orientational axis of rotation, and ϕ is amount of rotation For an unconstrained body, the translational and rotational equations of motion are given as follows:

$$m_i\ddot{r}_i = f_i \tag{14}$$

$$J_i\omega'_i + \tilde{\omega}'J_i\omega'_i = n'_i \tag{15}$$

where m_i, represent mass of the body "i", r_i, acceleration of the mass "i" after, J_i its inertia tensor; ω' and $\dot{\omega}'$ its angular velocity and acceleration respectively; f_i, n'_i are the forces and moments acting on the body "i" respectively. For a constrained system, such as the robot with its legs either attached to the surface or not, constraint equation Φ exist, which describes the dependency of coordinate q; i.e.,

$$\Phi = \Phi(q) = 0 \tag{16}$$

The differential equation of motion in the matrix form will be in the following form.

$$M\ddot{q} = g + \Phi_q^T\lambda \tag{17}$$

where

M is the system mass matrix;
q is the system acceleration vector;
g contains the applied and the gyroscopic forces and moments,
λ contains a set of Lagrange multipliers associated with the constraints,
$\Phi_q^T\lambda$ denotes the forces applied at the joints.

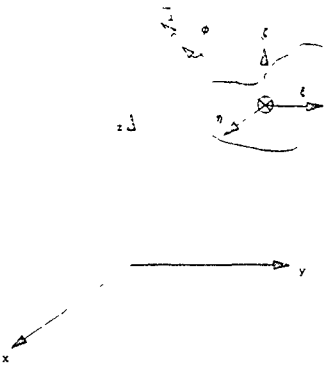


Figure 7 Configuration of body "i" with respect to xyz coordinate system.

Corresponding to these second-order differential equations of motion, twice as many initial conditions are needed on the coordinates q and the velocities $\dot{q}$ as.

$$q(0) = q^0, \quad q(0) = q.^0 \tag{18}$$

For these mixed differential-algebraic equations of motion for the robot were solved by a simple numerical algorithm The second time derivatives of the constraint equations were appended to the differential equations of motion to form a linear set, which at any instant can be solved for the system acceleration and Lagrange multipliers A direct integration method using the Cartesian coordinates and velocities as the state variable of integration, together with a forth-order Runge-Kuta with an adjustable step size were employed.

For a simple falling of the robot in a plane, with no other external forces other than gravity acting on the robot, the analysis reduces to a two dimensional analysis For other conditions, the 3D analysis must be performed Either way, the trajectory of the climbing robot can be obtained under the action of different loads. In order to visualize the dynamic behavior of the ROSTAM III during its fall, the result of the simulation has been converted to an animation program

In the analysis, the right leg was assumed to be in the extended form and the left leg in the retract form Three test cases were used to evaluate performance and establish rate requirements for the leg pivot motor All cases initialized the robot with zero rates and accelerations and the legs positioned at zero. Starting attitudes of the robots body were chosen as 90°, 120°, and 180°, and the result of animation are shown in Figures 8, 9, and 10 respectively. No cases less than 90° were chosen because it is assumed the

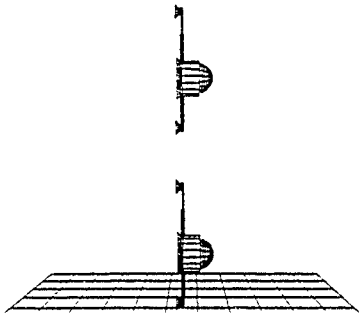


Figure 8 The simulation of the robot falling with a 90° angle

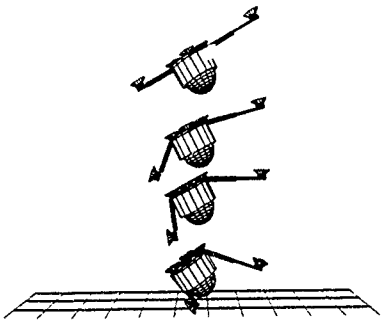


Figure 9 The simulation of the robot falling with a 120° angle.

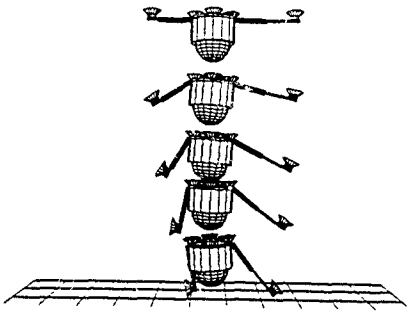


Figure 10 The simulation of the robot falling with a 180° angle

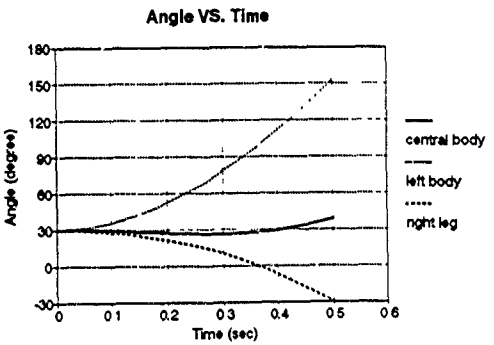


Figure 11 The angular position of the robot components with respect to inertial frame

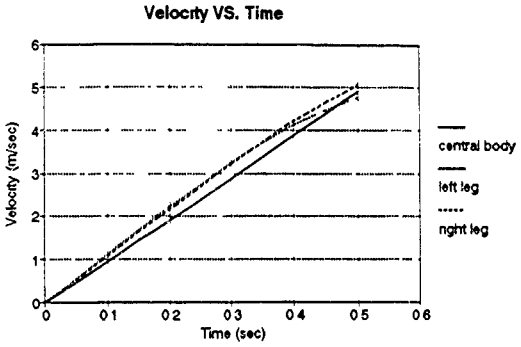


Figure 12 The velocity of the robot component with respect to inertial frame

robot would slide until an edge was reached. Minimum height that the system should respond in is initially taken to be 1 3 meters Final values for survivable height would be determined by how fragile the payload is. Figures 11, and 12 show the angular position and velocity of the each component of the robot during the fall

6. Conclusion

A description of ROSTAM III (Robotics System for Total Aircraft Maintenance) was presented in this paper which has two legs and a central rotational part for its movement. This robot was used to provide a prototype for the further needed research in the area of climbing robots It was shown that these robots can climb on any inclined plane, including vertical walls and ceilings. The safety analysis based on the assumption of the linear of the reaction force was developed for a general case, and the robot has proven to be safe for all conditions Suction cups were suggested for the adhesion of the robot to the surface, because they can stick to most type of the materials. Finally, in case the robot falls, a method was developed to save the robot and its expensive equipment. The results of this analysis show that the protection system is viable The animation and simulation program have proven to be very useful tool in evaluating the robot performance during the fall.

7. References

1 H Hemami, "Initiation of Walk and Tiptoe of Planar Nine-Link Biped", Mathematical Biosciences, Vol 61, pp 163-189, 1982

2. K J Waldron, "Force and Motion Management in legged Locomotion", IEEE, J Robotics Automation, Vol RA-2, No 4, D 1986

3. D A Messuri and C A Klein, "Automatic Body Regulation for Maintaining Stability of a legged Vehicle During Rough-Terrain Locomotion", IEEE, J Robotics and Automation, Vol RA-1, No.3, Sept, 1985

4 S M Song and K J Waldron, "Analytical Approach for Gait Study and its Applications on Wave Gaits", International Journal of Robotics Research, Vol.6, No.2, 1987

5. R B McGhee, A L Pai, "An Approach to Computer Control for Legged Vehicles", Journal of Terramechanics, Vol 11, No 1, pp.9-27, 1974.

6. C. Isik and A M Meystel, "Pilot Level of Hierarchical Controller for an Unmanned Mobile Robot", IEEE, J.Robotics and Automation, Vol 4, No 3, 1988.

7 J.L Crowley, "Navigation for Intelligent Mobile Robot", IEEE J Robotics and Automation, Vol.RA-1, No 1, March, 1985.

8 S Hirose, "Wall Climbing vehicle using Internally Balanced Magnetic Unit", Tokyo Institute of Technology Japan, 1987.

9. K. Sato, M. Watanabe, et "On Wall Travelling Robots for Nuclear Power Plants", Proceedings of the ANS Third Topical Meeting on Robotics and Remote Systems, Charleston South Carolina, March 13-16, 1989

10. N Akira, et al , "Design of a Robot Capable of Moving on a Vertical Wall", Advanced Robotics, Vol 1, pp.33 45.

11 N Akira, "Bipedal Walking Robot Capable of Moving on Vertical Wall for Inspection use", The 5th International Symposium on Robotics in Construction, June 6-8, 1988. Tokyo, Japan

12 " A Compendium of Robotics Equipment Used in Hazardous Environment", Electric Power Research Institute, EPRI NP-5697, pp.2-23, February, 1990.

13 Gradetsky V. G., Akselord, B V , et , "Motion of Climbing Robot With Manipulator Under Vibration", Robotics and Vision Conference, 1991.

14 S. Mann, "Design and Construction of a Wall-Climbing Robot", M.S Thesis, The Wichita State University, May, 1990

15. Z. Liu, "Design and Analysis for Surface-Climbing Robot", M S Thesis, The Wichita State University, 1991

16 B. Bahr, and F. Wu, "Design and Safety Analysis of a New Climbing Robot" Wichita State University, 1991

17. "Desktop Computer Animation a Guide to Low-cost Computer Animation", Gregory MacNicol, 1992

18. L Wen, B Bahr, and O Schwab, "Design of a Simulation System for Robot Motions," proceedings of the IASTED International Conference on Control and Robotics, vancouver Canada, August 4-6, 1992.

19 P E Nikravesh Computer-Aided Analysis of Mechanical System". Prentice-Hall, 1988
20 Strato Vacuum Handling Co . 4619 west Roscoe St Chicago, IL 60641, USA

Symbolic Computations in Flexible Multibody Systems

F. MELZER
Institute B of Mechanics
University of Stuttgart
W-7000 Stuttgart 80
Germany

Abstract The need for computer aided engineering in the analysis of machines and mechanisms led to a wide variety of general purpose programs for the dynamical analysis of multibody systems The use of more lightweight structures and an increasing demand of high-precision mechanisms, such as robots, led to the incorporation of flexible bodies in this methodology This paper presents a formalism for flexible multibody systems based on a minimum set of generalized coordinates and symbolic computation A standardized object oriented data model is used for the system matrices describing the elastodynamic behaviour of the flexible bodies. Consequently, the equations of motion are derived in a form independent of the chosen modelling technique for the elastic bodies

1 Introduction

Multibody dynamics evolved a quarter of a century ago from the need to simulate the dynamical behaviour of spacecrafts and machines. Nowadays, the method of multibody systems is well established and accepted, once it proved to be a valuable tool in the dynamical analysis of mechanisms such as robots, vehicles, and spacecrafts. Early works focused on the development of methodologies for rigid bodies only. The need to model as well the small structural deformations led to hybrid models. Likins [15] introduced hybrid coordinates to the multibody approach and the Finite Element community developed formulations to take into account the gross motion of the structural system, e g. Belytschko and Hsieh [4]. From these works the wide field of flexible multibody systems evolved. Presently, modelling techniques for flexible multibody systems can be classified into:

- Finite element approaches using a nonlinear formulation and absolute coordinates, e g Cardona and Geradin [6].
- Multibody approaches assuming a large gross motion and either small elastic deformations, e.g Shabana and Wehage [23], or nonlinear deformations due to geometric or material nonlinearities, e g. Ambrósio and Nikravesh [1].

This paper focuses on the multibody approach, which is very efficient from the computational point of view. Multibody models consist of rigid and flexible bodies interconnected by joints and force elements, as massless springs, dampers, and actively controlled elements, see fig 1 The joints have different properties and constrain the motion of the bodies

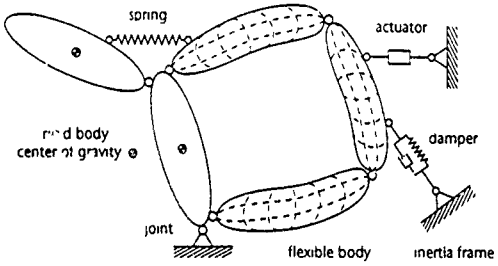


Fig 1 Schematic representation of a multibody system

In the multibody approach, techniques for the incorporation of flexible bodies are

- The superelement technique where a flexible body is described as a series of rigid bodies interconnected by elastic force elements, Rauh and Schiehlen [19].
- The continuum approach where a flexible body is discretized using either global or local shape functions, e.g Shabana and Wehage [23]. A variation of this approach is the lumped mass formulation, e.g. Kim and Haug [14]

With the availability of powerful general purpose computer algebra systems, such as *MAPLE* [7], *MATHEMATICA* [29], *AXIOM* [12], and *REDUCE* [5], symbolical computations are becoming more and more an important tool in engineering analysis Problems are rather solved in an analytical-numerical way than in a pure numerical way, see e g. the finite element approaches of Beltzer [3] and Ioakimidis [11]. The use of computer algebra may greatly reduce the effort for numerical calculations and facilitates the physical interpretation of the solution.

Symbolical formalisms for the dynamical analysis of rigid multibody systems are successfully used for more than a decade, Schiehlen and Kreuzer [21] The symbolical generation of the equations of motion is particularly attractive, since the equations of motion are derived in an explicit form, multiplications with zero elements are avoided, and the numerical procedure of time integration is decoupled from the derivation of the equations of motion, i e. the equations of motion have to be generated only once. Nevertheless, there are limitations in symbolic computing, in particular for flexible multibody systems. Drawbacks are, that only simple structural elements can be described by means of a symbolic approach for the displacement field, Beltzer [3]. Furthermore a large number of degrees of freedom may result in long expressions, which may be difficult to handle by a computer algebra system.

These drawbacks seem to be the reason that in a recent overview of available multibody codes, Schiehlen [22], none of the reported symbolic formalisms was able to deal with flexible bodies. There is, however, some recent work dealing with symbolical formalisms for flexible bodies, e g Ider and Amirouche [10] and Salecker [20], but so far they are restricted either to a certain class of bodies, mostly beams or trusses, or to planar problems.

The scope of this paper is to present a formulation of the equations of motion for flexible bodies undergoing a large gross motion and small elastic deformations overcoming above mentioned

drawbacks. A minimum set of generalized coordinates is used to generate the equations of motion in a compact form suitable for symbolic computation. A standardized object-oriented data base is used as an interface to numerically precomputed data describing the elastodynamical behaviour of a flexible body. The computer algebra system *MAPLE* is used for the generation of the equations of motion in terms of the objects of the data model. Since not all symbolical terms can be traced back to input variables, the equations are denominated semi-symbolical equations of motion.

2 Kinematics

The kinematic description of flexible and rigid bodies is done through the introduction of moving reference frames. The motion of a flexible body is described by the large nonlinear motion of its reference frame which is referred to as a "rigid"-body motion, and by small elastic deformations within this reference frame, see fig. 2.

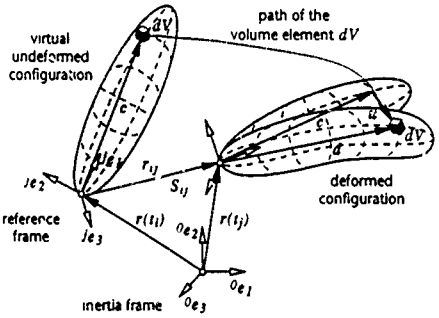


Fig. 2 General Motion of a free elastic body

KINEMATICS OF FLEXIBLE BODIES

The position and orientation of the reference frame j, formed by the vector basis ${}^{j}e = {}^{j}e_{\alpha}$ with $\alpha = 1(1)3$, relative to an inertial frame ${}^{0}e_{\alpha}$, $\alpha = 1(1)3$, is given by the translational vector r and the rotational tensor S, respectively. Denoting the location of a volume element dV in the undeformed configuration by c and the displacement field of the flexible body by $u = u(c, t)$, the position of the volume element dV with respect to the reference frame is expressed as

$$d(c, t) = c + u(c, t) \tag{1}$$

To arrive at a minimum number of degrees of freedom the displacement vector u is expressed as a linear combination of selected deformation modes,

$$u(c, t) = \Phi(c)q(t)\,. \tag{2}$$

where q are the generalized elastic coordinates. Methods for the selection of deformation modes within flexible multibody systems are presented by Kim and Haug [14] and by Meirovitch and Kwak [17]. Using the finite element method, the global shape functions Φ are written as

$$\Phi = S^T A \bar{S} B T . \tag{3}$$

where A is the element shape function matrix, S and $\bar{S}$ are the appropriate transformation matrices from the element to the reference frame, B is the Boolean matrix describing the assemblage of the finite elements, and T denotes the modal matrix. The deformation of the elastic body results in strains and a state of stress within the volume element dV. Strains are measured by the Green-Lagrange strain tensor whose components are written as

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial c_j} + \frac{\partial u_j}{\partial c_i} + \frac{\partial u_k}{\partial c_i}\frac{\partial u_k}{\partial c_j}\right) = \varepsilon_{ji} . \tag{4}$$

The strain vector

$$\varepsilon = [\varepsilon_{11} \;\; \varepsilon_{22} \;\; \varepsilon_{33} \;\; 2\varepsilon_{12} \;\; 2\varepsilon_{23} \;\; 2\varepsilon_{31}]^T = \varepsilon_L + \varepsilon_{NL} \tag{5}$$

is decomposed into a term ε_L, linearly dependent on the displacements u, and a nonlinear term ε_{NL}. It has to be pointed out that due to the large displacement of the flexible body the second order term ε_{NL} has to be considered in eq. (5). Using appropriate differential operator matrices A and $\bar{A}(u)$, eq. (5) is rewritten as

$$\varepsilon = Au + \bar{A}(u)u . \tag{6}$$

Substituting the displacement field by eq. (2) yields

$$\varepsilon = Lq + \bar{L}(q)q \tag{7}$$

where L and $\bar{L}(q)$ are the linear and the nonlinear strain matrices respectively.

The second Piola-Kirchhoff stress tensor is the energetic conjugate measure to the Green-Lagrange strain tensor. It is related to the strain tensor by the constitutive equation

$$\sigma_{ij} = H_{ijkl}\varepsilon_{kl} + \sigma^n_{ij} . \tag{8}$$

Here H_{ijkl} represents the material constitutive tensor and σ^n_{ij} is the initial or nominal state of stress due to the gross body motion.

KINEMATICS OF FLEXIBLE MULTIBODY SYSTEMS

Kinematic relations derived so far describe a single flexible body undergoing a large "rigid" body motion. A closer look at the topological description of a system of rigid and flexible bodies is now necessary. For the example in fig. 3, the position and orientation of the frame j is described by

$$r_j = r_i + r_{ij} , \tag{9}$$

$$S_j = S_i S_{ij} \tag{10}$$

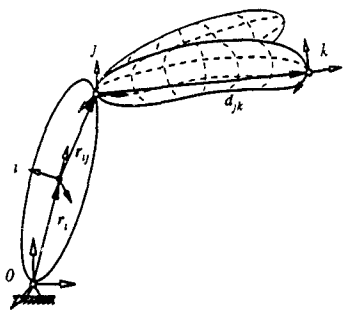


Fig. 3 Simple multibody system consisting of a rigid and a flexible body

where r_{ij} and S_{ij} denote the translational vector and the rotational tensor from the coordinate frame i to the frame j. Similarly, the motion of the coordinate system k attached to the elastic body is given by the following recursive formula

$$r_k = r_j + d_{jk} . \tag{11}$$

$$S_k = S_j S_{jk} . \tag{12}$$

In the following only elastic bodies will be considered. Rigid bodies can be treated as a special case of elastic bodies where the displacement field u is zero. Differentiating eqs (11) and (12) with respect to time yields the absolute velocities and the absolute accelerations as

$$v_k = r_k = v_j + \tilde{\omega}_j d_{jk} + {}_j u_k . \tag{13}$$

$$\omega_k = \omega_j + \omega_{jk} . \tag{14}$$

$$a_k = \dot{r}_k = a_j + \dot{\tilde{\omega}}_j d_{jk} + 2\tilde{\omega}_j {}_j u_k + \tilde{\omega}_j \tilde{\omega}_j d_{jk} + {}_j \dot{u}_k , \tag{15}$$

$$\alpha_k = \dot{\omega}_k = \alpha_j + \alpha_{jk} + \tilde{\omega}_j \omega_{jk} \tag{16}$$

where the left subscript denotes the coordinate frame to which the time derivation is referred to. Subscripts or superscripts denoting the inertia frame are generally omitted. The skew symmetric tensor $\tilde{\omega}$ of the angular velocity is found as

$$\tilde{\omega}_j = \dot{S}_j S_j^T . \tag{17}$$

Consider now a multibody system of p rigid or elastic bodies and h holonomic constraints. The system holds f degrees of freedom

$$f = 6p - h + n_q \,, \tag{18}$$

and is uniquely described by $6p - h$ generalized coordinates y, describing the "rigid"-body motion of the flexible bodies, and by n_q generalized elastic coordinates q describing the elastic deformations. Let the vector of the generalized coordinates, denoted by y_q, be written as,

$$y_q = \begin{bmatrix} y \\ q \end{bmatrix} \tag{19}$$

For a flexible multibody system the absolute translational accelerations a_j and rotational accelerations α_j, in terms of the generalized coordinates, are:

$$a_j = \frac{\partial r_j}{\partial y_q^T} \dot{y}_q + \frac{\partial v_j}{\partial y_q^T} y_q + \frac{\partial v_j}{\partial t} = J_{Tj}(y_q, t)\dot{y}_q + \bar{a}_j(y_q, \dot{y}_q, t) \,, \tag{20}$$

$$\alpha_j = \frac{\partial s_j}{\partial y_q^T} \dot{y}_q + \frac{\partial \omega_j}{\partial y_q^T} y_q + \frac{\partial \omega_j}{\partial t} = J_{Rj}(y_q, t)\dot{y}_q + \bar{\alpha}_j(y_q, \dot{y}_q, t) \tag{21}$$

Applying eqs. (20) and (21) to eq. (15) the acceleration may now be expressed in terms of the generalized coordinates y_q as

$$a_k = J_{Tj}\dot{y}_q - \tilde{d}_{jk}J_{Rj}\dot{y}_q + \bar{a}_j - \tilde{d}_{jk}\bar{\alpha}_j + 2\tilde{\omega}_{j\,j}u_k + \tilde{\omega}_j\tilde{\omega}_j d_{jk} + {}_j u_k \,, \tag{22}$$

where the $3 \times f$ Jacobian matrices are denoted by J_T and J_R and the vectors of the local accelerations are denoted by $\bar{a}$ and $\bar{\alpha}$.

3 Dynamics

The equations of motion for a multibody system with flexible members are for a holonomic system derived by D'Alembert's principle:

$$\sum_{j=1}^{p} \left(\int_V \delta r^T (\rho a - f^a)\, dV + \int_V \delta \varepsilon^T \sigma^a dV - \int_A \delta r^T f^a dA \right)_j = 0 \,. \tag{23}$$

The virtual work of the constraint forces vanishes

$$\delta r^T f^c = 0 \tag{24}$$

Therefore, only applied forces f^a have to be considered in eq. (23).

LINEARIZATION OF THE GENERALIZED ELASTIC COORDINATES

Up to this point the formulation is nonlinear. The question remaining, therefore, is how far it is necessary to use a nonlinear description to arrive at an accurate and computational efficient model. Linearization of the elastic deformations within the reference frame, i.e. assuming only small

deformations around the undeformed configuration, is a common approach to gain computational efficiency, e g Shabana [24] Focusing on the virtual work of the internal forces, eq. (23) is written as

$$\sum_{j=1}^{p} \left(\int_V \delta q^T (L + \bar{L}(q))^T (HLq + \sigma^n) \, dV = R \right)_j \tag{25}$$

with

$$R = \int_A \delta r^T t^a dA - \int_V \delta r^T (\rho a - f^a) \, dV \tag{26}$$

Neglecting terms of second order in q, eq (25) is written in a more compact form as

$$\sum_{j=1}^{p} [\delta q^T (K_L q + K_{NL}(q) + f^\sigma) = R]_j \tag{27}$$

The matrices K_L and K_{NL} denote the linear stiffness matrix and the geometric stiffness matrix, respectively. The generalized forces related to the actual state of stress are denoted by the vector f^σ:

$$K_L = \int_V L^T HL dV , \tag{28}$$

$$K_{NL} = \int_V \bar{L}^T(q) \sigma^n dV , \tag{29}$$

$$f^\sigma = \int_V L^T \sigma^n dV \tag{30}$$

The equations given above represent the principle of virtual work for a multibody system with flexible bodies, linearized in the generalized elastic coordinates. It is important to note, that K_{NL} represents a first order term in the dynamical equations based on kinematical terms of second order, i e the strain matrix $\bar{L}(q)$.

TIME-INVARIANT SYSTEM MATRICES

Evaluating all the volume integrals in eq. (27) is a lengthy process. In order to reduce the computational burden during simulation, the integrals are expressed by generalized elastic coordinates and by system matrices which are time-invariant, and, therefore, precomputable. The inertia tensor of a flexible body serves as an example to illustrate this process:

$$I(q) = -\int_V \rho \tilde{d} \tilde{d} dV = -\int_V \rho \tilde{c} \tilde{c} dV - \int_V \rho \tilde{c} \tilde{u} dV - \int_V \rho \tilde{u} \tilde{c} dV - \int_V \rho \tilde{u} \tilde{u} dV . \tag{31}$$

Using eqs. (2) and (3) the inertia tensor can be also written in index notation as

$$I_{gh} = I^0_{gh} + C_{gnh}q_n + \underbrace{D_{gnmh}q_n q_m}_{\approx 0} \tag{32}$$

with the time-invariant matrices I^0_{gh} and C_{gnh}, where I^0_{gh} represents the constant inertia tensor of the rigid body, and

$$C_{gnh} = (\varepsilon_{fgd}\varepsilon_{djh} + \varepsilon_{fhd}\varepsilon_{djg})\, T_{ln} \sum_i c^i_f S^i_{ej} \tilde{S}^i_{rk} B^i_{kl} \int_{V^i} \rho A^i_{er} dV^i \tag{33}$$

Table 1 presents all volume integrals of eq (27), the corresponding expression in index notation, and the corresponding time-invariant system matrices. The corresponding integrals of the element shape functions for a Bernoulli beam element were derived by Shabana [24]

volume integral	index notation	FE-formulation of the time-invariant system matrix
$\int_V \rho u dV$	$C^1_{gn}q_n$	$C^1_{gn} = T_{ln}\sum_i S^i_{eg}\tilde{S}^i_{rk}B^i_{kl}\int_{V^i}\rho A^i_{er}dV^i$
$\int_V \rho\tilde{c}\Phi dV$	C^2_{hn}	$C^2_{hn} = \varepsilon_{hfj}T_{ln}\sum_i c^i_f S^i_{ej}\tilde{S}^i_{rk}B^i_{kl}\int_{V^i}\rho A^i_{er}dV^i$
$\int_V \rho\Phi^T\Phi dV$	C^3_{pn}	$C^3_{pn} = T_{op}T_{ln}\sum_i \tilde{S}^i_{sm}\tilde{S}^i_{rk}B^i_{mo}B^i_{kl}\int_{V^i}\rho A^i_{es}A^i_{er}dV^i$
$\int_V \rho\tilde{c}\tilde{u}dV$	$C^4_{gnh}q_n$	$C^4_{gnh} = \varepsilon_{gfd}\varepsilon_{djh}T_{ln}\sum_i c^i_f S^i_{ej}\tilde{S}^i_{rk}B^i_{kl}\int_{V^i}\rho A^i_{er}dV^i$
$\int_V \rho\Phi^T\tilde{\omega}\Phi dV$	$C^5_{pnj}\omega_j$	$C^5_{pnj} = \varepsilon_{hjg}T_{op}T_{ln}\sum_i B^i_{mo}B^i_{kl}S^i_{dh}S^i_{eg}\tilde{S}^i_{sm}\tilde{S}^i_{rk}\int_{V^i}\rho A^i_{ds}A^i_{er}dV^i$
$\int_V \rho\Phi^T\tilde{\omega}\tilde{\omega}udV$	$C^6_{fpnj}\omega_f\omega_j q_n$	$C^6_{fpnj} = \varepsilon_{hfd}\varepsilon_{djg}T_{op}T_{ln}\sum_i B^i_{mo}B^i_{kl}S^i_{dh}S^i_{eg}\tilde{S}^i_{sm}\tilde{S}^i_{rk}\int_{V^i}\rho A^i_{ds}A^i_{er}dV^i$

Table 1 Time-invariant system matrices

STIFFNESS PROPERTIES

In the following the integrals of eqs (28) to (30) are considered in more detail The evaluation of the linear stiffness matrix K_L is straight forward. Assuming linear elasticity, K_L is derived from eq. (28) in index notation as

$$K_{L_{pn}} = T_{op}T_{ln}\sum_i B^i_{mo}B^i_{kl}\tilde{S}^i_{sm}\tilde{S}^i_{rk}\int_{V^i}\rho\Lambda^i_{db}A^i_{ds}ff^i_{bc}\Lambda^i_{ce}A^i_{er}dV^i \tag{34}$$

The issue of the significance of the geometric stiffness or stress stiffness matrix, especially in high-speed mechanisms, was addressed by various researchers, e g Likins [16] and Kane et al [13]. Different methods for the computation of the stress stiffness matrix have been developed, e g Ider and Amirouche [10] proposed an explicit stiffness matrix, which has to be computed at

every time step A more efficient approach is the superposition method as proposed by Wallrapp, Santos and Ryu [26][26] and by Banerjee and Dickens [2][2]. Eq (29) indicates that the geometric stiffness matrix is proportional to the state of nominal stress which again is proportional to the loads. Assuming the linear theory of elasticity, unit stress stiffness matrices for different load cases can be precomputed The stress stiffening matrix is found as the sum of the individual unit matrices scaled with the actual load at time integration. To evaluate σ^n the undeformed configuration has to be analysed, i.e. $q = \dot{q} = \ddot{q} = 0$ For this configuration the state of nominal stress is quasi-statically equilibrated with the inertia loads and the applied external loads, and σ^n is computed from eq. (27) for any time t.

An efficient formulation of the superposition method is obtained using the properties of the Bernoulli beam Assuming an one-dimensional state of stress and neglecting shear, the consistent element geometric stiffness matrix for a Bernoulli beam element has the form

$$G^i_{mk} = \int_l A^i_{hm,1}\, A^i_{hk,1}\, dx \quad , \qquad A^i_{hm,1} = \frac{\partial A^i_{hm}}{\partial x} \quad . \tag{35}$$

In table 2, the individual stress stiffness matrices are shown for a structure made out of Bernoulli beam elements with length l and cross section F. The total length of this straight structure is L.

load case	ind. stress stiffness matrix K_{NL}	volume integral	FE-formulation of the time-invanant system matrix for a straight Bernoulli beam
longitudinal force f_1	$\frac{1}{2}\Phi(q)f_1$	–	$\Phi = -\sum_i T^T B'^T G^i B^i T q$
translational acceleration	$C^7(q)J_T\ddot{y}_q$	$C^7(q) = \int_V \Phi(q)dV$	$C^i = \sum_i \frac{1}{2}\rho F T^T B'^T \int_l G^i(L - c_i)\,dx B^i T q$
rotational acceleration	$C^8(q)J_T\ddot{y}_q$	$C^8(q) = \int_V \tilde{c}\,\Phi(q)dV$	$C^8 = 0$
centrifugal acceleration	$C^9(q)\tilde{\omega}\tilde{\omega}$	$C^9(q) = \int_V c\,\Phi(q)dV$	$C^9 = \sum_i \frac{1}{4}\rho F T^T B'^T \int_l G^i(L^2 - c_i^2)\,dx B^i T q$

Table 2 Individual stress stiffness matrices for a straight Bernoulli beam

It was pointed out, that kinematic terms of second order are necessary in order to describe the geometric stiffness correctly By comparison, the term $0,5\,\Phi(q)$ is identified as the one to modify the global shape function in second order. Terms representing the geometric stiffness are then incorporated in the mass matrix and in the vector of the gyroscopic forces.

STANDARD INPUT DATA

A current issue in the development of software tools for the computer aided analysis of mechanical systems is the data transfer between different programs or modules Within the German research project *Dynamics of Multibody Systems* an object-oriented data model has been developed by Otter et al. [18] This allows not only the communication between different modules of a complex software package, but defines also a clear interface to other programs. Wallrapp [28]

extended this object-oriented data model to flexible bodies. The basic idea is to collect the time-invariant system matrices at a very high abstraction level in a data model, see table 3.

Through a Taylor expansion of an object M_{ij}, e g. a time-invariant system matrix, in the form

$$M_{ij}(q_n) = M^0_{ij} + M^1_{inj} q_n + M^2_{inpj} q_n q_p + . \tag{36}$$

linear or nonlinear models of the flexible body, derived either by continuum or by finite element methods, may be incorporated in the database. Due to the object-oriented approach the user has access to such objects as the structure of a matrix expansion. This is particularly attractive for symbolic computing. Without knowing the values of the matrix elements, the structure of a matrix, e g. diagonal or unit matrix, is accessible in the data model. The use of such objects may considerably reduce the overhead for the symbolical generation of the equations of motion. All Terms describing the elastodynamic behaviour of a flexible body are incorporated in this data model.

$\mathcal{M}(q)$	description	dimension	computation rule:	$M_{ij}(q_n) = M^0_{ij} + M^1_{inj} q_n$
$m\tilde{d}_s$	matrix of mass times positionof centre of mass	3 x 1	$M^0 = \int_V \rho c dV$	$M^1_{gn} = C^1_{gn}$
I	inertia tensor	3 x 3	$M^0 = -\int_V \rho \tilde{c}\tilde{c} dV$	$M^1_{gnh} = -C^4_{gnh} - C^4_{hng}$
C_T	coupling matrix with rotational coordinates	n_q x 3	$M^0 = C^{1T}$	$M^1_{gnp} = C^7_{gnp}$
C_R	coupling matrix with rotational coordinates	n_q x 3	$M^0 = C^{2T}$	$M^1_{pnj} = C^5_{pnj}$
M_E	mass matrix of flexible body	n_q x n_q	$M^0 = C^3$	
G_R	gyroscopic matrix for rotational coordinates	3 x n_q x 3	$M^0_{gz} = -2C^4_{gnh}$	
G_E	gyroscopic matrix for elastic coordinates	n_q x n_q x 3	$M^0_{pz} = 2C^5_{pnh}$	
O_E	centrifugal matrix for elastic coordinates	3 x n_q x 3	$M^0_{px} = C^4_{hpg}$	$M^1_{pnx} = C^6_{fpnj} + C^9_{fpnj}$
K_E	stiffness matrix of flexible body	n_q x n_q	$M^0 = K_L$	
D_E	damping matrix of flexible body	n_q x n_q	$M^0 = \alpha C^3 + \beta K_L$	

Table 3 Computing some of the classes of the standard input data using a FE-approach

Equations of motion for flexible multibody systems

Based on the data model the equations of motion can be formulated as

$$M(y_q)\,\dot{y}_q(t) + k_c(y_q, y_q) + k_f(q, q) = q_f(y_q, y_q)\,, \tag{37}$$

with the symmetric mass matrix

$$M = \sum_{j=1}^{p} \Big(J_T^T m J_T - J_T^T m \tilde{d}_s(q) J_R + J_T^T C_T^T(q) J_E + J_R^T m \tilde{d}_s(q) J_T + J_R^T I(q) J_R$$
$$J_R^T C_R^T(q) J_E + J_E^T C_T(q) J_T + J_E^T C_R(q) J_R + J_E^T M_{EE} J_E \Big)_j \quad . \tag{38}$$

the vector of the generalized gyroscopic forces,

$$k_c = \sum_{j=1}^{p} \Big(J_T^T m \tilde{a} - J_T^T m \tilde{d}_s(q) \bar{\alpha} + J_T^T \tilde{\omega} \tilde{\omega} m d_s(q) + 2 J_T^T \tilde{\omega} C_T^T(q) \dot{q} + J_R^T m \tilde{d}_s(q) \bar{\alpha} + J_R^T I(q) \bar{\alpha} +$$
$$J_R^T \tilde{\omega} I(q) \omega + J_R^T G_R(q) \omega q + J_E^T C_T(q) \bar{a} + J_E^T C_R(q) \bar{\alpha} + J_E^T O_E(q) \Omega + J_E^T G_L(q) \omega q \Big)_j \tag{39}$$

the vector of the internal forces,

$$k_I = \sum_{j=1}^{p} \Big(J_E^T K_E q + J_E^T D_E \dot{q} \Big)_j \, , \tag{40}$$

and the vector of the generalized forces

$$q_f = \sum_{j=1}^{p} \Big(J_T^T m g + J_R^T m \tilde{d}_s(q) g + J_E^T C_T(q) g \Big)_j \; +$$
$$\sum_{j=1}^{p} \sum_{n} \Big(J_T^T f_e^n + J_R^T \tilde{d}^n(q) f_e^n + J_R^T l_e^n + J_E^T \Phi^{nT}(q) f_e^n + J_E^T \Psi^{nT}(q) l_e^n \Big)_j \, . \tag{41}$$

In these equations the matrix J_E represents the Jacobian of the generalized elastic coordinates

$$J_E = \frac{\partial q_j}{\partial y_q^T} \tag{42}$$

which describes how the vector q_j is related to the vector y_q

4 Generating semi-symbolical equations of motion

The presented formalism is implemented in a general computer program using the symbolical formalism *NEWEUL* and the computer algebra system *MAPLE*. The former is a specialized computer algebra system designed for the derivation of the equations of motion for mechanical systems with rigid bodies only. *NEWEUL* has a restricted set of available symbolic operations and is based on index coding, Schiehlen and Kreuzer [21]. *MAPLE* is a multipurpose computer algebra program for numerical and symbolical computations. It is also a programming language supporting procedural programming, Char et al. [7]. Compared to other computer algebra systems, *MAPLE* uses the required memory in a very economical way. Together with the possibility to generate complete *FORTRAN* and *C* programs, *MAPLE* is well suited for the development of prototype programs. For this purpose the lack of computational speed, compared to specialized computer algebra systems, is acceptable.

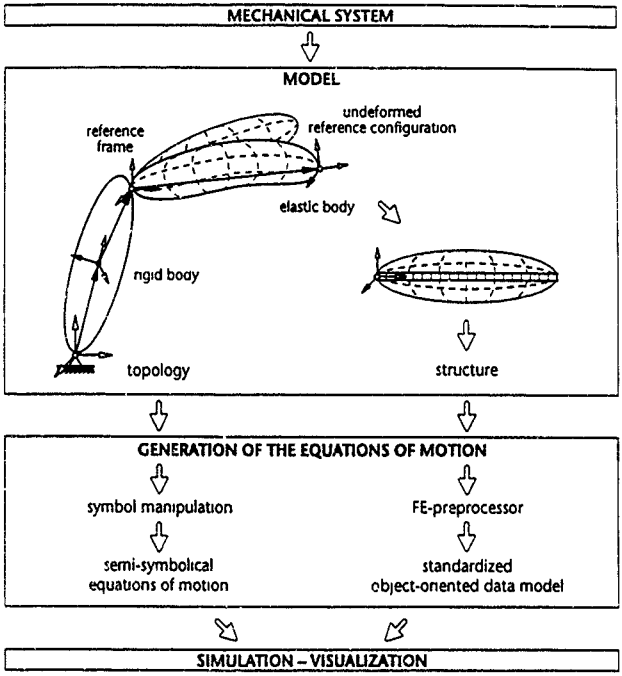


Fig 4 Scheme of a dynamical analysis of flexible mechanical systems

Based on the kinematical relations of the flexible mechanical system generated by *NEWEUL* the equations of motion are generated in a semi-symbolical form using *MAPLE* as a symbolical machine The terminology *semi-symbolical* was used, since not all symbolical terms can be traced back to input variables. Such terms are for example the classes of the data model. To avoid unnecessary operations, the ob ct *structure of a matrix* is utilized to define the classes of the data model in such a way that zero-operations are eliminated as soon as possible. Semi-symbolical terms can not be interpreted easily in a physical way, but they lead to a computationally efficient formulation overcoming the restriction to certain types of models for the elastic body In order to keep the expressions in a compact form, repeated strings of symbols have to be replaced by substitution variables This can be realized either by assigning these strings to an arbitrary substitution variable

or by replacing them by physically interpretable symbols. A example for such a physical interpretable symbol is the expression of the third component of the ω vector by O1(3). A very compact symbolical form of the equations of motion is achieved by using the recursive kinematical formulas and replacing the strings recursively. This procedure is referred to as recursive compression.

A scheme of a dynamical analysis of flexible mechanical systems is depicted in fig. 4. The equations of motion are generated by symbol manipulation based on the topological description of the system. Independently a FE-preprocessor, currently based on Bernoulli beam elements, computes numerically the time-invariant system matrices and stores them in the data model. These matrices are read into in the semi-symbolical equations as a parameter set and the output is a ready-to-compile FORTRAN code. The ordinary differential equations are then integrated using standard algorithms, see e.g. Gear [9]. It has to be noted that the modules *symbol manipulation*, *FE-preprocessor*, and *time integration* are independent, connected only by a clearly defined data model, see fig. 4.

5 Examples

Two examples of flexible mechanisms will be presented: the well known example of the rotating beam and a flexible robot.

ROTATING BEAM

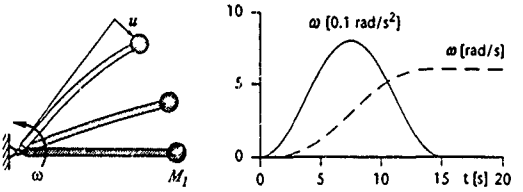


Fig. 5 Rotating beam and imposed angular velocity and acceleration

A slender beam, length 10 m, with an additional tip mass M_1 is rotating around a hinge with a given function of the angular acceleration ω. It represents a spin-up manoeuvre starting at time $t = 0$ s and ending at $T = 15$ s reaching at a constant angular velocity $\omega_s = 6$ rad/s, see fig. 5. The function for the corresponding angle $\varphi(t)$ is given as:

$$\varphi(t) = \begin{cases} \frac{\omega_s}{T}\left(\frac{t^2}{2} + \left(\frac{T}{2\pi}\right)^2 \left(\cos\frac{2\pi t}{T} - 1\right)\right) & \text{for } t < T \\ \omega_s\left(t - \frac{T}{2}\right) & \text{for } t \geq T \end{cases} \tag{43}$$

No gravity acts on the beam and the properties are as follows. cross section area $A = 0.0004\ m^2$, area moment of inertia $I_3 = 2x10^{-7}\ m^4$, Youngs's modulus $E = 7x10^{10}\ N/m^2$, and mass density $\rho = 3000\ kg/m^3$, Ref [10].

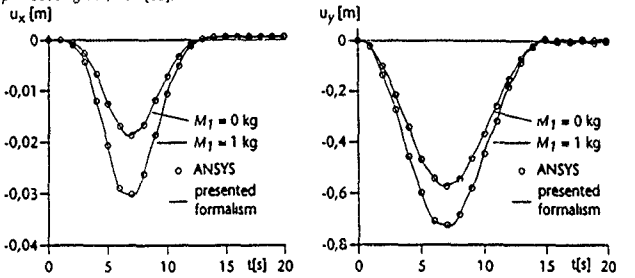


Fig. 6 Elastic tip displacements u_x and u_y of the rotating beam

Mass matrix in symmetric storage mode:

```
M(1) = LT2(1,2)**2*M1+LT2(2,2)**2*M1+ME_e1(1,1)
M(2) = LT2(1,2)*LT2(1,3)*M1+LT2(2,2)*LT2(2,3)*M1+ME_e1(1,2)
M(3) = LT2(1,3)**2*M1+LT2(2,3)**2*M1+ME_e1(2,2)
M(4) = LT2(1,2)*LT2(1,4)*M1+LT2(2,2)*LT2(2,4)*M1+ME_e1(1,3)
M(5) = LT2(1,3)*LT2(1,4)*M1+LT2(2,3)*LT2(2,4)*M1+ME_e1(2,3)
M(6) = LT2(1,4)**2*M1+LT2(2,4)**2*M1+ME_e1(3,3)
```

Right hand side.

```
qe(1) = -LT2(1,2)*AQT2(1)*M1-LT2(2,2)*AQT2(2)*M1-
        OE_e1(1,3)*O1(3)**2-GE_qp_e1(1,3)*O1(3)-KE_e1(1,1)*q_e1_1-
        (LT2(1,1)*LT2(1,2)*M1+LT2(2,1)*LT2(2,2)*M1+CR_e1(1,3))*A1PP
qe(2) = -LT2(1,3)*AQT2(1)*M1-LT2(2,3)*AQT2(2)*M1-
        OE_e1(2,3)*O1(3)**2-GE_qp_e1(2,3)*O1(3)-KE_e1(2,2)*q_e1_2-
        (LT2(1,1)*LT2(1,3)*M1+LT2(2,1)*LT2(2,3)*M1+CR_e1(2,3))*A1PP
qe(3) = -LT2(1,4)*AQT2(1)*M1-LT2(2,4)*AQT2(2)*M1-
        OE_e1(3,3)*O1(3)**2-GE_qp_e1(3,3)*O1(3)-KE_e1(3,3)*q_e1_2-
        (LT2(1,1)*LT2(1,4)*M1+LT2(2,1)*LT2(2,4)*M1+CR_e1(3,3))*A1PP
```

Notation.

q_e1_1: $q(1)$ O1: ω A1PP: $\dot{\omega}$ LT2: J_{T2} AQT2: $\tilde{a}_2$ _e1: elastic body 1

Table 4 Semi-symbolical equations of motion for the rotating beam

The time history of the elastic displacements u_x and u_y of the tip with respect to the undeformed state of the beam are shown in fig 4 and are compared to the solution obtained with the finite-element code *ANSYS* [8] A very good agreement of the solutions was obtained with the presented formalism The reduction of the model resulted in a strong increase in computational efficiency Modelling with *ANSYS* was done using 10 2D-beam elements resulting in a system of sparse dif-

ferential equations of dimension 33 For the presented example 2 bending modes and 1 longitudinal mode were selected leading to a very compact system of ordinary differential equations of dimension 3 The equations of motion are presented in fig 4 They have a very compact form and with the object definitions of table 3 they can be easily interpreted. Additional equations for the taylor expansion of the data objects, eq. (36), have to be generated. They are omitted in this paper due to the limited space

FLEXIBLE ROBOT

A planar flexible robot was presented as a benchmark example within the German research project *Dynamics of Multibody Systems*, see Sorge, Bremer and Pfeiffer[25]. Gravity with $g = 9.81\ m/s^2$ acts on the robot and material properties are chosen as Youngs's modulus $E = 7{,}3 \times 10^{10}\ N/m^2$, and mass density $\rho = 2700\ kg/m^3$ The pointmasses are selected as $M_1 = 1\ kg$ and $M_2 = 3\ kg$. Dimensions of the undeformed beams are for beam I: $x = 0{,}545\ m$, $y = 0{,}015\ m$, $z = 0{,}06\ m$ and for beam II: $x = 0{,}675\ m$, $y = 0{,}01\ m$, $z = 0{,}04\ m$ The functions of the joint angles φ_1 and φ_2 are give . as

$$\varphi_1 = \begin{cases} -\pi/4 & \text{for } t < 0 \\ \pi/4\,(-1+72t^2) & \text{for } 0 \le t < 1/6 \\ \pi/4\,(-18t+108t^2-144t^3) & \text{for } 1/6 \le t < 1/3 \\ \pi/4\,(-8+54t-108t^2+72t^3) & \text{for } 1/3 \le t < 1/2 \\ \pi/4 & \text{for } 1/2 \le t \end{cases} \tag{44}$$

$$\varphi_2 = -\varphi_2 \tag{45}$$

Three distinct points have to be calculated, the equilibrium position for $t = 0\ s$, and the two extreme points of the vibration around the equilibrium position for $t > 0.5\ s$, see fig. 7

reference solution			presented formalism		
time [s]	x [m]	y [m]	time [s]	x [m]	y [m]
0	1.057	-0 411	0	1 057	-0 410
0 41	1 019	0 599	0 42	1.019	0 592
0 57	1 043	0 123	0 57	1 046	0 123

Table 5 Selected positions for the robot

Table 5 indicates a good agreement of the solution obtained by the presented approach with the reference solution, see Sorge, Bremer and Pfeiffer[25]

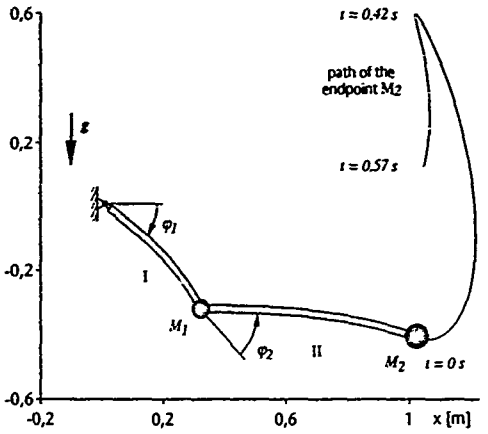


Fig. 7 Flexible robot

6 Conclusion

Based on the topological description of a multibody system with flexible members the equations of motion were derived in a minimal form using D'Alembert's principle. System matrices describing the elastodynamical behaviour were derived for a finite element approach and the geometric stiffness matrix is given for the case of a straight beam. By use of a standardized object oriented data base it was shown that the presented formalism can be implemented in a semi-symbolical form, without restrictions on the type of model used for the elastic body. The two presented examples, a rotating beam and a flexible robot, indicated a good agreement with reference solutions

References

[1] Ambrósio, J. A. C., Nikravesh, P. E., *Elasto-Plastic Deformations in Multibody Dynamics*. Nonl Dynamics 3 (1992) pp. 85 - 104.

[2] Banerjee, A. K.; Dickens, J. M.: *Dynamics of an Arbitrary Flexible Body in Large Rotation and Translation*. AIAA J. Guidance 13 (2) (1990) pp. 221 - 227.

[3] Beltzer, A. I., *Variational and Finite Element Methods: A Symbolic Computation Approach*. Berlin,... Springer, 1990

[4] Belytschko, T.; Hsieh, B. J.: *Nonlinear Transient Finite Element Analysis with Convected Coordinates* Int. J. Num. Meth. Eng. 7 (1973) pp. 255 - 271.

[5] Brackx, F.; Constales, D., *Computer Algebra with LISP and REDUCE* Dordrecht,...: Kluwer, 1991.

[6] Cardona, A.; Geradin, M.: *A Beam Finite Element Non-linear Theory with Finite Rotations*. Int. J. Num. Meth. Eng. 26 (11) (1988) pp. 2402 - 2438.
[7] Char, B. W. et al.: *First Leaves: Tutorial Introduction to Maple*. New York: Springer, 1992.
[8] DeSalvo, J. G.; Gorman, R. W.: *ANSYS Engineering Analysis System User's Manual*. Houston: Swanson Analysis Systems, 1989.
[9] Gear, C. W.: *Numerical Initial Value Problems in Ordinary Differential Equations*. Englewood Cliffs: Prentice Hall, 1971.
[10] Ider, S. K.; Amirouche, F. M. L.: *Nonlinear Modelling of Flexible Multibody Systems Dynamics Subject to Variable Constraints*. J. of Appl. Mech. 56 (1989) pp. 444 - 450.
[11] Ioakimidis, N. T.: *Elementary Applications of MATHEMATICA to the Solution of Elasticity Problems by the Finite Element Method*. Comp. Meth. Appl. Mech. Eng. 102 (1993) pp. 29 - 40.
[12] Jenks, R. D.; Sutor, R. S.: *AXIOM. The Scientific Computation System*. Berlin...: Springer, 1992.
[13] Kane, T. R.; Ryan, R. R.; Banerjee, A. K.: *Dynamics of a Cantilever Beam Attached to a Moving Base*. J. Guid. 10 (2) (1987), pp. 139 - 151.
[14] Kim, S. S.; Haug, E. J.: *Selection of Deformation Modes for Flexible Multibody Dynamics*, Mech. Struct. & Mach., 18 (4) (1990) pp. 565 - 586.
[15] Likins, P. W.: *Finite Elements Appendage Equations for Hybrid Coordinate Dynamic Analysis*. Int. J. Solids Struct. 8 (1972) pp. 709 - 731.
[16] Likins, P. W.: *Geometric Stiffness Characteristics of a Rotating Elastic Appendage*. Int. J. Solids Struct. 10 (1974) pp. 161 - 167.
[17] Meirovitch, L.; Kwak, M. K.: *Convergence of the Classical Raleigh-Ritz Method and the Finite Element Method*. AIAA-J. 28 (8) (1990) pp. 1509 - 1516.
[18] Otter, M.; Hocke, M.; Daberkow, A.; Lemke, G.: *Ein objektorientiertes Datenmodell zur Beschreibung von Mehrkörpersystemen unter Verwendung von RSYST*, Institutsbericht IB-16. Stuttgart: Universität, Institut B für Mechanik, 1990.
[19] Rauh, J ; Schiehlen, W.: *Various Approaches for the Modelling of Flexible Robot Arms*. In: Elishakoff, I.; Irrener, H. (Eds.): Refined Dynamical Theories of Beams, Plates and Shells and their Applications. Berlin: Springer, 1987, pp. 420 - 429
[20] Salecker, M.: *Zur Dynamik hybrider Mehrkörpersysteme - Theorie und symbolische Programmierung*. Ph. D dissertation, Faculty of mechanical engineering, University of Karlsruhe, 1991.
[21] Schiehlen, W. O ; Kreuzer, E. J.: *Symbolic Computerized Derivation of Equations of Motion* In: Magnus, K. (Ed.): Dynamics of Multibody Systems, IUTAM-Symposium München, Germany. Berlin...:Springer, 1976
[22] Schiehlen, W. (Ed.): *Multibody Systems Handbook*. Berlin: Springer, 1989.
[23] Shabana A. A.; Wehage, R.: *Variable Degree-of-freedom Component Mode Analysis of Inertia Variant Flexible Mechanical Systems* J. of Mech., Transm., and Autom. in Design 105 (1983) pp. 371- 378.
[24] Shabanna, A. A.: *Dynamics of Multibody Systems*. New York: J. Wiley, 1989.
[25] Sorge, K.; Bremer, H.; Pfeiffer, F.: *Multi-Body Systems with Rigid-Elastic Subsystems*. In: Schiehlen, W (Ed.) Advanced Multibody System Dynamics - Simulation and Software Tools. Dordrecht: Kluwer Academic Publishers, 1993, pp. 195 - 215.
[26] Wallrapp, O.; Santos, J.; Ryu J.: *Superposition Method for Stress Stiffening in Flexible Multibody Dynamics*. In. Kirk, C. L; Junkins, J. L. (Eds): Proc. Int. Conf. on Dynamics of Flexible Structures in Space. Cranfield, UK. Berlin: Springer, 1990, pp 233 - 247.
[27] Wallrapp, O ; Schwertassek, R.: *Representation of Geometric Stiffening in Multibody System Simulation* Int. J. Num. Meth. Eng. 32 (1991) pp. 1833 - 1850
[28] Wallrapp, O.: *Standard Input Data of Flexible Members in Multibody Systems*. In Schiehlen, W. (Ed.) Advanced Multibody System Dynamics - Simulation and Software Tools Dordrecht: Kluwer Academic Publishers, 1993, pp 445 - 450.
[29] Wolfram, S.. *Mathematica: A System for Doing Mathematics by Computer* Reading: Addison-Wesley, 1988

Flexible Multibody Dynamics — An Object-Oriented Approach

Martin Anantharaman
Fachgebiet Mechatronik, Universität —GH— Duisburg
Lotharstr. 1 4100 Duisburg Germany

April 1993

Abstract

An approach to the computer aided analysis of flexible multibody systems using object-oriented programming methods is presented. The aim is to support the rapid generation of specialized programs by providing an open, extensible C++ toolkit. This toolkit contains modules (C++ classes) which allow the declaration and manipulation of multibody components such as joints, bodies and actuators in an intuitive manner. New components (e.g. new finite elements) are easily introduced to extend the toolkit. The equations of motion for a multibody system consisting of these components are formulated by direct application of the principle of virtual work using symbolic techniques. It is possible to use absolute as well as relative coordinates in a problem-dependent manner.

1 Introduction

Methods for the computer-aided analysis of multibody systems have reached a high level of maturity and are being increasingly applied to real-world problems in various sectors of industry. Powerful programs such as ADAMS, DADS, MECANO and COMPAMM have established themselves as standard tools within a CAE environment and are being rapidly enhanced to meet user needs.

While such programs, with sophisticated graphical user interfaces and links to other analysis tools and CAD programs, are well suited to the industrial context, research in multibody methods also requires a kind of programming environment in which rapid experimentation is possible. Such experimentation, which might involve using completely different methods for formulating equations, comparing different formulations for a new component or testing new numerical solution procedures, is not easy with existing programs and conventional programming techniques. A

possible alternative lies in the use of *object-oriented programming* to create *toolkits* from which researchers can rapidly generate customized programs.

One such approach to applying object-oriented programming techniques to the analysis of multibody systems (see [Sorge, 1992, Kecskeméthy, 1993] for other recent work) will be presented in the following. The primary aim is to create a framework within which different types of coordinates and various forms of equations may be experimented with. No attempt is made to formulate a multi-purpose program, but as will be shown, the resulting toolkit allows formulation of specialized simulation programs at the same level of abstraction as most input languages for multibody programs. The theoretical background is described in Section 2 and fundamental aspects in the design of the toolkit are discussed in Section 3. A key element in the present approach is the combination of symbolical and numerical computations in a set of classes for mathematical quantities introduced in Section 4. Based on these mathematical classes one can define abstractions for the elements of multibody systems as described in Section 5. The assembly of a complete multibody system model from such elements is demonstrated in Section 6 and the application of various analysis procedures to such a model is shown in Section 7.

2 Theoretical background

A distinctive feature of multibody dynamics is that no single method has established itself in the same way as the finite element method in structural dynamics. In the following, the major current approaches or "formalisms" will be briefly reviewed, focussing on the type of coordinates used and the computational tasks arising from assembly of the dynamical equations of motion[1].

In the *absolute-coordinate* or *cartesian-coordinate approach* (see e.g. [Nikravesh, 1987]), a translation vector $\mathbf{r}$ and some rotation parameters Θ are introduced to independently describe the motion of one or more reference frames on each body and are collected in the vector of generalized coordinates $\boldsymbol{q}$ of the multibody system. The system's vector of generalized forces $\boldsymbol{Q}$ is composed of the translational force $\boldsymbol{f}$ and a torque $\boldsymbol{t}$ applied to the various reference frames. Here and in the following, we assume that forces also include inertial forces unless otherwise stated. The generalized forces are easily formulated by additively accumulating contributions from dynamic elements such as bodies, springs etc. Joints are modelled with constraint equations, which are simply concatenated to form the multibody system's constraint equations $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$. Dynamic equilibrium is defined by equating the generalized forces $\boldsymbol{Q}$ with constraint forces $\boldsymbol{G}^T\boldsymbol{\lambda}$, which are formulated with the help of Lagrange multipliers $\boldsymbol{\lambda}$ and the constraint Jacobian $\boldsymbol{G}$. For the solution of the equations arising in kinematical and dynamical analysis one also requires the mass and stiffness matrices and the

[1] The so-called "recursive" methods, such as those of [Featherstone, 1987, Bae and Haug, 1987] are not considered here, as their unique combination of equation formulation and equation solution put them in a category of their own.

Jacobian of the constraints in sparse form. All these operations can be performed numerically and are easy to automatize, so that this is the method of choice for most large commercial multibody program packages.

To avoid the large number of coordinates and constraint equations arising in the absolute-coordinate approach, the *relative-coordinate* or *joint-coordinate approach* (see e g. [Wittenburg, 1977]) uses natural joint coordinates as the generalized coordinates q. The motion of reference frames is now expressed in terms of these joint coordinates by formulating the functions $r(q)$ and $R(q)$ for translation and rotation. Formulation of the generalized forces Q is more involved than in the absolute-coordinate approach, as a force or torque applied to a reference frame may make a contribution to any number of components of the generalized forces. Numerous methods have been developed for this purpose, based on diverse mechanical principles such as Lagrange's equations of the second kind, Kane's equations or the principle of D'Alembert. In all these methods the crucial step is to accumulate projections $J_r^T f$ and $J_R^T t$ of forces and torques acting on reference frames, using Jacobians or "velocity transformations" J_r and J_R of the translation and rotation of these frames Kinematical loops are handled by setting up constraint equations in the joint coordinates, which are concatenated to form the multibody system's constraint equations $g(q) = 0$. Dynamic equilibrium is defined in the same way as for the absolute-coordinate method by equating the generalized forces with constraint forces. The implementation of the joint-coordinate approach requires some sort of topological analysis of the system to identify kinematical loops and formulate the motion of all reference frames. During kinematic and dynamic analysis, first the position and velocity of all reference frames has to be evaluated, followed by computation of absolute forces and constraint equations. The accumulation of generalized forces then proceeds according to specialized algorithms generally involving recursive evaluation of the Jacobians J_r and J_R or recursive application of their transpose, (see [Kecskeméthy, 1993] for a review). Evaluation of the mass matrix and the constraint Jacobian is quite complex and very few algorithms permit evaluation of stiffness matrices

The *minimal-coordinate approach* of [Hiller and Kecskeméthy, 1989] is a form of the joint-coordinate approach specialized to systems with multiple kinematical loops, in which all constraints arising from the kinematical loops are solved in closed form for one or more joint coordinates in each loop. Only the remaining joint coordinates are included in the vector of generalized coordinates q. Since no constraint equations or Lagrange-multipliers occur, the number of coordinates and equations is minimal, but the formulation of the closed-form solutions further adds to the complexity inherent in joint-coordinate approaches. A fully automatic implementation of the method is extremely challenging, but a semi-automatic implementation in the form of a C++ toolkit is described in [Kecskeméthy, 1993].

Each of these three approaches has its advantages and disadvantages, so that a *combined approach* seems promising, in which a suitable combination of different coordinate-types can be used to take advantage of the special structure of each

multibody system. Such a combined approach may be developed as an extension of the joint-coordinate approach, as the underlying mechanical principles in general permit the use of any other type of coordinates, including absolute coordinates. New aspects arising in such an extension of the joint-coordinate approach are the special treatment of finite rotations (which rarely occur as joint coordinates), generalization of algorithms for the efficient evaluation of generalized forces and exploitation of sparsity. The present approach attempts this extension of the joint-coordinate method and has the following distinctive features:

- Coordinates of any kind are allowed, e.g. joint coordinates, cartesian coordinates, strain measures etc., but a distinction is made between a "position" and "velocity" representation to support the use of parameters of finite rotations (see Section 5.1).
- Dynamic elements are formulated in a coordinate-independent fashion to supply the virtual work of a generalized applied force (see Section 5.3), e.g. a translational force, torque or even a stress measure performing virtual work against a corresponding virtual strain measure.
- Assembly of a multibody system and its equations is based on the application of symbolic techniques (see Section 3 and 4), which obviates the need for specialized algorithms to attain efficiency and also permits exploitation of sparsity.
- Computational procedures for the analysis of the multibody system (see Section 7) are based on a special model of numerical procedures (see Section 4) which permits different analysis procedures to be combined.

3 Design of the multibody toolkit

In this section the basic design of a toolkit for analysis of multibody systems in the object-oriented programming language C++ [Stroustrup, 1991] will be discussed. A discussion of object-oriented programming as such and its advantages with respect to other programming paradigms is not possible here (see e.g. [Stroustrup, 1988, Booch, 1991]), also a familiarity with the basic concepts of C++ is assumed.

Object-oriented programming is, far more than conventional procedural programming a design task and the key element of "object-oriented design" [Booch, 1991] is the identification of classes as abstractions of entities in the problem domain and the definition of the behaviour of these classes. Here, the problem domain is the analysis of multibody systems and it is natural to try to encapsulate components of multibody systems such as joints, bodies, forces and constraints as classes. In general (see [Sorge, 1992, Kecskeméthy, 1993]) this involves defining a lower level of classes for standard mathematical objects such as scalars, vectors and matrices and

characterizing the behaviour of components of multibody systems by certain typical computations with such mathematical objects Considering the different formalisms discussed in Section 2 it is immediately obvious that different formalisms require different computations, e g. in the absolute-coordinate approach a joint must compute its constraint equations, while in the joint-coordinate approach it must be able to compute the motion of the outboard reference frame.

We may say that *different formalisms imply different abstractions* and this applies to an even greater extent to the treatment of the multibody system as a whole, where every formalism requires its own global data structures, leading to completely different representations of a multibody system as a class. The absolute-coordinate approach essentially requires simple mappings between quantities such as state variables and forces in the components of the multibody system and the corresponding composite quantities in the multibody system While the order in which computation are performed is of little importance in the absolute-coordinate approach, the joint-coordinate approach requires recursive computation of the motion of reference frames and of the Jacobians or velocity transformations, generally involving some form of "tree" or "graph" data structure.

Trying to encapsulate such diverse abstractions of multibody systems as a collection of unrelated classes is a tedious task unless one recognizes that the different concepts have much in common: one is always dealing with *scalar, vector and matrix functions and their derivatives.* It should also be kept in mind that such standard numerical procedures as solution of linear and nonlinear equations and numerical integration require some representation of vector-valued functions and their partial derivatives, preferably as sparse matrix-valued functions.

Such considerations have led to the development of a C++ toolkit for multibody systems in which the main emphasis is on the design and implementation of mathematical operations This resulting toolkit has the following structure:

- A module for symbolical and numerical mathematics defines differentiable scalar, vector and matrix functions. Such functions can be defined as expressions in a syntax very close to standard mathematical notation and an important feature is the assembly of composite vectors or "hypervectors" to represent product spaces.
- General numerical procedures are defined in an abstract sense in terms of such functions, exploiting the ability to automatically formulate exact derivatives of scalars and vectors as vector and matrix functions (as sparse block-oriented matrices or "hypermatrices" in the case of hypervectors), respectively
- Kinematic elements such as joints and constraints recursively formulate functions of state variables using the facilities for expression formation.
- Dynamic elements formulate the virtual work of generalized forces against virtual displacements as a real-valued functional and the constraints formulate a virtual work of constraint forces.

- A multibody system assembles vectors such as state variables and constraint functions and accumulates virtual work from its components Generalized forces are formulated by partial differentiation of the virtual work and system matrices such as the mass matrix, stiffness matrix an * constraint Jacobians are formulated by partial differentiation of the generalized forces and the constraint functio..

- Analysis procedures for multibody systems are simply derived from classes for general numerical procedures and essentially extract the appropriate function to solve and the vector to solve for from the multibody system.

Further details on these components of the toolkit will be given in the following sections.

4 Numerical and symbolical computations

The classes for scalar, vector and matrix operations mentioned in the previous section will be introduced "by example", focussing on the classes for vector functions derived from the base class[2] `Rn`, but the design of classes for scalar functions and matrix functions (derived from `R1` and `Rmxn`) is similar. `Rn` is an abstract base class specifying properties common to all vector functions (or simply "vectors") and subclasses specialize `Rn` to specific vector functions. In the simplest case we have the subclass `RVec` to represent independent vectors which can be declared just like an integer or real number with statements like

```
RVec x (3, 0); // x = (0, 0, 0)
RVec y (x);
RVec z (x);
```

Vector functions are generally the result of *expression formation* as in

```
Rn& a = x + y;
Rn& b = x % y;
```

for the sum and cross product of vectors. Here, the application of the operator "+" to two vectors internally generates an object of the "expression class" `RnPlusRn`. It must be stressed that this sum object is not the (constant) *value* of the sum at the point of declaration but represents a *function* mapping the values of the two operands to the value of the sum at any time. For this `RnPlusRn` and other expression classes store references to the operands for later use. Note that no numerical operations are performed at the time of expression formation: numerical operations are performed during subsequent *evaluation* of the function, e.g. when its value is required for evaluation of another expression or if it is to be printed as in

[2]Typewriter font will be used in the following for program symbols and code

```
cout << a;
```

An important feature of evaluation is *value caching*, i.e. on subsequent evaluations a stored ("cached") value is used until an operand changes in value. Thus, if immediately after the evaluation due to printing shown above we assign

```
z = a; // Assignment only!
```

the sum is *not reevaluated*, but if we assign to one of its operands as in

```
x = 5;
```

the expression a is *notified* of this change and

```
z = a; // Evaluation followed by assignment!
```

forces a reevaluation. Value caching is exploited by the mechanism of *common subexpression elimination*, which guarantees that whenever a subexpression equivalent to an existing one is formed, the new ("common") subexpression is eliminated and the existing subexpression is reused. Thus

```
Rn& c = z % (x + y);
```

reuses a and on evaluation of c, the cached value of a is used. In addition, certain *expression simplifications* are built into expression formation, e.g. the cross product of a vector with itself creates a "zero vector" or ZeroRn and use of this ZeroRn in subsequent expressions results in its elimination.

Expression formation can proceed recursively to any level of complexity, leading to the formation of large numbers of expression objects linked to one another in the form of "expression trees". By virtue of expression simplification, common subexpression elimination and value caching, these expression trees automatically achieve most of the optimizations typically found in specialized multibody algorithms, so that the need for such algorithms practically vanishes.

Another important operation is *symbolic differentiation*, which allows the formulation of exact derivatives of a vector function with respect to one of its operands as in

```
Rmxn& da_x = a.d (x);
```

and makes the formulation of multibody system matrices such as Jacobians, mass matrices and stiffness matrices by hand unnecessary.

While expression formation and differentiation as described till now is appropriate for small to medium vectors and implicitly assumes full matrices, the class HyperRn defines "hypervectors" composed of elementary vectors and HyperRmxn represents "hypermatrices", which are sparse matrices with elementary matrices as elements. HyperRns are used in the formation of product spaces from elementary vector functions as in

```
HyperRn q;
q.append (x);
q.append (y);
```

but are also generated as the gradient of scalar functions with respect to a HyperRn as in

```
R1& s = ...;
HyperRn& ds_x = s.d (q);
```

Similarly, HyperRnxns may be manually constructed but are more often created automatically as the derivative of an HyperRn with respect to another HyperRn as in

```
HyperRn f;
f.append (a);
f.append (b);
HyperRnxn& df_q = f.d (q);
```

These operations are the basis for the assembly of multibody systems from their components (see Section 6)

It is interesting to compare this object-oriented implementation of basic mathematical operations with programming environments for *numerical* mathematics such as Matlab or MATRIX$_X$ and systems for *symbolical* mathematics like Mathematica and Maple. Clearly, the toolkit implemented here does not attain the power of these commerical programs in most areas, but the peculiar combination of symbolical and numerical operations is particularly suited to multibody system analysis Thus, the ability to form partial derivatives and easily create composite vectors and matrices is required for the assembly of multibody systems and the combination of common subexpression elimination with value caching to enable efficient numerical evaluation was motivated by well-known algorithms to compute generalized forces and Jacobians.

A further benefit of this synthesis of symbolical and numerical mathematics becomes obvious in the object-oriented model of *numerical solution procedures* adopted here: a numerical solution procedure or "solver" is considered to be a *constraint* on an existing independent vector, and forces this vector to behave as the solution function of some numerical problem. An abstract base class RnSolver implements the basic mechanism involving close cooperation between the solver object and the solution vector, and a "solver class" derived from RnSolver defines a particular solution method for a certain numerical problem. For instance, the solution of nonlinear equations by Newton's method is implemented in the class RnNewton and the solution of equations f = 0 for the vector q may be declared with

```
RnNewton q_solver (q, f),
```

The solver object q_solver cannot be manipulated or accessed in any way, but as long as it exists the actual iterative process to form the solution in q will be automatically activated whenever q is evaluated, e g. when we print it with

```
cout << q,
```

A modified form of value caching is implemented to recompute the solution only when some parameter influencing `f` changes. In addition, one may formulate the sensitivity of the solution with respect to such a parameter by differentiating the constrained vector `q`. Numerical procedures implemented in this style include the solution of linear equations and numerical integration of ordinary differential equations and differential-algebraic equations.

5 Elements of multibody systems

In the following subsections the formulation of various components of multibody systems based on the mathematical operations introduced in the previous section will be described. First, *state variables* are introduced, which are used in *kinematic elements* to formulate kinematical quantities such as the motion of reference frames. Based on the kinematics, *dynamic elements* formulate the virtual work of various kinds of forces and *constraints* formulate constraint equations.

5.1 State variables

State variables represent generalized coordinates that determine the state of a multibody system. They consist of the following components: position variables $\bar{p}$, velocity variables v, their time derivatives p and $\dot{v}$, virtual displacements δp and velocity functions $p'(p, v)$, $v'(p, \dot{p})$ to transform between $\dot{p}$ and v. In the following, we will use the symbol q for such state variables.

Distinguishing between position and velocity variables simplifies the definition of state variables to describe rotations, as we can use any parametrization (e.g. Euler-parameters) for the position variables while using angular velocity for the velocity variables. In this case a transformation is required between angular velocity and the time derivatives of the Euler-parameters and for this reason the velocity functions p' and v' are introduced. The virtual displacements δp are required for the formulation of virtual work in dynamic elements and are independent vectors in the same space as the velocity variables. Thus, e.g. the virtual displacement of a state variable describing a finite rotation is a differential rotation vector corresponding to the angular velocity.

The implementation of state variables is straightforward: the class `State` has components for p, v, $\dot{p}$, v, δp, $\dot{p}'$ and v', but to support composition of state variables, these components are `HyperRn`s, which must be assembled by subclasses of `State`. Composition of state variables is implemented in `State` in the obvious way as concatenation of the `HyperRn` components.

5.2 Kinematic elements

State variables are used in *kinematic elements* to formulate functions describing the motion of the multibody system. A central role is played by *rigid-body motion* between *reference frames*. Within the present approach, a reference frame (or simply "frame") is directly identified with a rigid-body motion. This motion may be relative to an inertial reference frame or any other reference frame, so that it may describe an absolute or relative motion, depending on the context. The rigid body motion is described as usual by a translation r and a rotation R, translational and angular velocity v and ω and translational and angular acceleration a and α, further, we introduce translational and angular virtual displacements δr and δR. It is assumed that all these vectors are represented in some unspecified global coordinate system, but we also consider a local or body-fixed coordinate system rotated by the frame's rotation R and sometimes use representations in this coordinate system.

A frame is represented by the class `Frame` which stores a reference to an `Rmtx` for R and references to `Rns` corresponding to r, v, ω, a, δr and δR represented in the global and body-fixed coordinate system. `Frame` defines "multiplication" of `Frame`s for the result of two successive motions and "transposition" for reversed motions. For given `Frame`s `X_1` and `X_2` we could thus write

```
Frame X_3 = X_1 * X_2; // X_2 followed by X_1
Frame X_4 = ~X_1;      // Inverse of X_1
```

While frames represent rigid-body motions in their most general form, it is generally more convenient to form rigid-body motions from specialized elementary frames which formulate subgroups of rigid-body motion in terms of joint coordinates. Such elementary frames will be called "links" and are represented by subclasses of `Frame` such as `RevLink` for the rotation about an axis and `PrismLink` for the translation along an axis.

While neither frames nor links are associated with any state variable, a "joint" is considered here to be a link with an associated state variable used as joint coordinate. Thus, e.g. a `RevJoint` derived from a `RevLink` includes a state variable of the dimension 1 and `Node` which is directly derived from `Frame` uses a state variable of dimension 6 to formulate a general rigid-body motion.

5.3 Dynamic elements

Dynamic elements are those components of a multibody system which generate applied forces. In the present approach we demand that dynamic elements actually formulate a virtual work δW as a real-valued functional. This not only enables the projection of applied forces onto generalized forces in a simple manner (see Section 6) but also accomodates non-standard types of forces such as stress measures Typically, formulation of virtual work will make use of kinematic elements such as the

frame of the centre of mass of a rigid body or the frames of the ends of a beam element. The virtual work will also generally involve parameters specific to the dynamic element such as the mass and inertia tensor of a rigid body.

The class `Dynamic` is the base class for representations of dynamic elements. It stores references to `Rls` for the virtual work δW and for kinetic and potential energy K and U, which are provided for diagnostic purposes. Simple subclasses of `Dynamic` are `Force` and `Torque` for a single applied force and torque, `RigidBody` for rigid bodies, `LinSpring` and `LinTorsion` for translational and rotational spring-damper actuators. More complex classes are `Beam` for a nonlinear beam element based on [Simo, 1985] and `SuperElement` for a superelement comparable to those described in [Cardona and Géradin, 1991]. The implementation of `Beam` will be presented in the oral presentation to illustrate the ease with which new dynamic elements can be introduced.

5.4 Constraints

The joints of a multibody system may also be modelled as *constraints*. This is necessary when the system contains kinematical loops, but may even be beneficial for systems without loops, to reduce the coupling within the system model (see Section 6). Constraints place restrictions on state variables by formulating *constraint functions* which have to be zero. The constraint functions consist of position constraint functions g, velocity constraint functions $\dot{g}$, acceleration constraint functions $\ddot{g}$ and virtual displacement constraint functions δg and are generally formulated with respect to two reference frames. A Lagrange-multiplier λ is associated with each constraint and a virtual work of the constraint forces is formulated as $-\delta g \cdot \lambda$ In the following we will write g for the position constraint function or the whole constraint, depending on context.

The components of `Constraint` must be assembled in subclasses and such subclasses of `Constraint` have been defined for all standard joint types. These subclasses include elementary constraints such as `SpherConstraint` for spherical joints as well as composite constraints such as `RevConstraint`, which models the constraints of a revolute joint by composition of a `SpherConstraint` and two `DotConstraints` (see e g. [Nikravesh, 1987]). For this purpose, `Constraint` defines a composition of constraints similar to the composition of `States`.

In most cases, constraints will be used to take into account kinematical loops, but in some cases it is possible to solve such loop-closure constraints for the joint-coordinates of particular joints in the loop. Several such closed-form solutions, based on the method of [Woernle, 1988], have been implemented here in the form of special links. These "solver links" may be viewed as kinematically driven joints, the joint-coordinates being driven in such a way that the loop-closure constraints are identically satisfied. E.g. a `LengthRevSolver` is a kind of revolute joint driven in such a way that a length-constraint is satisfied.

6 Assembly of multibody systems

A multibody system is considered to have its own state variables q, virtual work δW and constraint g. Thus, a multibody system is treated as a combination of state variables, dynamic elements and constraints and in fact the class `System` is derived from `State`, `Dynamic` and `Constraint`. The multibody system is thought to start out empty, without state variables or constraints, and a zero virtual work. One then adds state variables (which are appended to q), dynamic elements (which are added to δW) and constraints (which are appended to g *and* added to δW). In the class `System` these operations are easily defined in terms of the composition operations already defined in `State`, `Dynamic` and `Constraint`.

The main purpose of `System` is to formulate various vector and matrix functions required by analysis procedures. The most important of these are the generalized forces, now termed f. Their formulation is based on the fact that the virtual work δW is known to be a linear functional of the form

$$\delta W(\dot{v}, v, p, \lambda, \delta p) = f(\dot{v}, v, p, \lambda) \cdot \delta p. \tag{1}$$

While the generalized forces f in this functional are not available in explicit form, they can be easily extracted with the help of the following partial differentiation:

$$f = \frac{\partial \delta W}{\partial \delta p} \tag{2}$$

Within `System` this operation is performed by symbolic differentiation of the `R1` δW with respect to the `HyperRn` δp and by virtue of the optimizations implemented in expression formation, the resulting `HyperRn` f will allow efficient numerical evaluation for any kind of state variable.

The generalized forces f depend on the acceleration $\dot{v}$, velocity v, position p, and, if the multibody system model contains constraints, the Lagrange-multipliers λ. Various analysis procedures require partial derivatives of the generalized forces, and `System` formulates the mass matrix M, damping matrix D and stiffness matrix K as `HyperRnxns` by symbolic differentiation of f according to

$$M = \frac{\partial f}{\partial \dot{v}}, \; D = \frac{\partial f}{\partial v}, \; K = \frac{\partial f}{\partial p}. \tag{3}$$

Similarly, the Jacobian matrice of the constraint equations are required. Here we assume that

$$\frac{\partial \dot{g}}{\partial v} = \frac{\partial \ddot{g}}{\partial \dot{v}} = \frac{\partial \delta g}{\partial \delta p} \tag{4}$$

and therefore `System` formulates only

$$G_p = \frac{\partial g}{\partial p}, \; G_v = \frac{\partial g}{\partial v} \tag{5}$$

Finally, partial derivatives of the velocity functions (see Section 5.1)

$$P = \frac{\partial p'}{\partial v}, \; V = \frac{\partial v'}{\partial \dot{p}} \tag{6}$$

are provided.

An important feature of the current approach is that the formulation of generalized forces and other functions required by analysis procedures (see Equation (2) to Equation (6)) is independent of the type of coordinates used, once the multibody system has been assembled. Note also that many of the quantities formulated here are not available in existing approaches, e.g. the stiffness and damping matrices are almost never available in closed form in the current implementations of the joint-coordinate approach

For a given mechanical system, assembly of a multibody system model within the present approach actually proceeds in several steps.

1. Definition of links and joints. Most state variables will also be created in this step as part of joints or nodes.
2. Formulation of the kinematical structure, if necessary, by constructing frames from chains of links or joints. If a closed-form solution of loops is to be attempted, it has to be formulated at this stage.
3. Definition of constraints on the kinematical structure.
4. Definition of dynamic elements based on the kinematical structure.
5. Addition of selected state variables, dynamic elements and constraints to the multibody system model.

In general, considerable freedom exists in the first three steps, which define the kinematics On the other hand, the definition of dynamic elements is generally independent of the coordinates used. By adopting a suitable strategy for defining kinematics it is possible to arrive at models with different kinds of coordinates corresponding to the well-known multibody formalisms mentioned in Section 2. This is briefly described in the following:

Absolute coordinates: For each body we introduce a `Node` which describes a general rigid-body motion with six degrees of freedom, but all joints are modelled as constraints After adding all the nodes and constraints to the multibody system, the system will have a large number of coordinates and constraints On the other hand, the system matrices will be quite sparse, and this sparsity is preserved in the `HyperRmxns` formulated here for the system matrices.

Joint coordinates: We must analyze the system topology to detect loops and partition the joints into "tree joints" and "cut joints". The tree joints are modelled with joint classes whereas the cut joints are modelled as constraints. The joints and constraints are added to the system which will have a medium number of coordinates and constraints (if there were any loops).

Minimal coordinates: We must again detect kinematical loops, but now we must partition the joints in each loop into "independent joints" and "dependent joints". The independent joints are modelled with joint classes while the dependent joints are represented by solver links encapsulating closed-form solutions of the loop-closure constraints. Only the independent joints are included in the multibody system model, so that the number of coordinates is minimal.

Apart from these basic strategies, combinations are also possible, e.g we might want to solve certain loops in closed from, but selectively use constraints elsewhere to uncouple the system

As an example, consider the simple four-bar mechanism of Figure 1, which is now to be modelled according to the strategies outlined above. In each case, the model will consist of declarations of C++ objects from the classes introduced in this section and the previous sections. In particular, heavy use will be made of expressions of **Frames** (see Section 5.2) to formulate kinematic chains. An inertial reference frame and certain links corresponding to rigid connections will be used in all cases and are declared first:

```
Frame X_0; // Inertial reference frame

PrismLink d (2, 5.0); // Translate along z-axis by 5.0 m
PrismLink a (0, 10.0); // Translate along x-axis by 10.0 m
PrismLink b (1, -10.0);
PrismLink c (2, 2.0),
R t_length = 14.456832; // The length of the coupler
PrismLink t (0, -t_length);
PrismLink half_t (0, -0.5*t_length);
```

Four-bar with absolute coordinates: We declare nodes for each of the three moving bodies, and then formulate constraints for the four joints with the following code

```
Node N_1; // Bottom of d
Node N_2; // Bottom of c
Node N_3; // Right end of t

Frame X_m = N_3 * half_t; // Centre of t
```

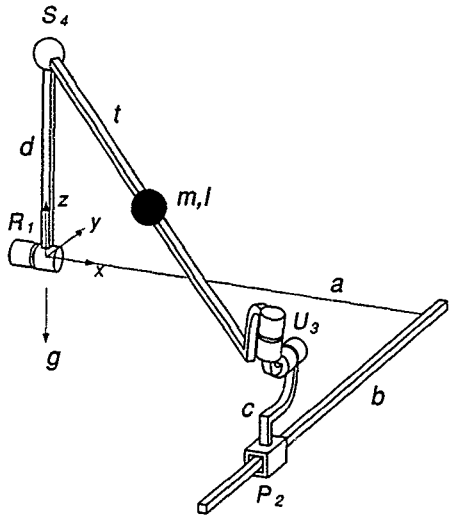


Figure 1. Four-bar mechanism.

```
RevConstraint R_1 (X_0, N_1, 0);
PrismConstraint P_2 (X_0 * a * b, N_2, 1);
UnivConstraint U_3 (N_2 * c, N_3, 1, 2);
SpherConstraint S_4 (N_3 * t, N_1 * d);

System system;
system.add (N_1);
system.add (N_2);
system.add (N_3);
system.add (R_1);
system.add (P_2);
system.add (U_3);
system.add (S_4);
```

Four-bar with joint coordinates: We cut the loop at the spherical joint S_4, which is modelled as a constraint, and declare R_1, P_2 and U_3 as joints in the following manner:

```
RevJoint R_1 (0);
```

```
PrismJoint P_2 (1),
UnivJoint U_3 (1, 2);

Frame X_3 = X_0 * a * b * P_2 * c * U_3; // Right end of t
Frame X_m = X_3 * half_t;

SpherConstraint S_4 (X_0 * R_1 * d, X_3 * t),

System system;
system.add (R_1),
system.add (P_2),
system.add (U_3);
system.add (S_4);
```

Four-bar with minimal coordinates: We again cut the loop at S_4, but the loop-closure constraint is decomposed into a length-constraint and a parallel-constraint by considering the characteristic pair $S_4 - U_3$ as described in [Woernle, 1988] The length constraint is solved to determine the translation within the prismatic joint P_2 and the parallel-constraint is solved to determine the angles within the universal joint U_3. Therefore, only R_1 is declared as an independent joint, while solver links are formulated for P_2 and U_3 using additional chains of links.

```
RevJoint R_1 (0);

Frame X_1 = ~b * ~a * R_1 * d; // Left half of bottom segment
LengthPrismSolver P_2 (X_1, c, t, 1, true);

Frame X_b = ~c * ~P_2 * X_1; // Bottom segment
ParUnivSolver U_3 (X_r, t, 1, 2);

Frame X_m = X_0 * a * b * P_2 * c * U_3 * half_t;

System system;
system.add (R_1);
```

For the sake of simplicity only one dynamic element, a concentrated mass attached to the centre of the coupler, is considered. It was ignored in the models above, but a frame X_m was formulated for the centre of mass In all cases the four-bar can be completed in the following manner

```
m = 10.0;
I = m*t_length*t_length/12.0,
```

```
PrismLink g (2, 0.0, 0.0, 9.81); // Gravitational acceleration
RigidBody body (X_m, g, 10.0);
body.I(1,1) = body.I(2,2) = I;

system.add (body);
```

At this stage, the models of the four-bar are complete and analysis procedures can be applied to them, as will be shown in the following section. While the declaration of the models is quite simple and intuitive, it is important to realize that the declarations are fragments of C++ code that could be compiled and linked to executable programs.

7 Analysis of multibody systems

Various analysis procedures can be applied to the models of multibody systems assembled in the previous section. The implementation of the analysis procedures is in the form of "multibody solver classes" derived from general solver classes introduced in Section 4, which implement general numerical solution procedures. A multibody solver specializes a general solver by applying the solution procedure to equations formulated from a `System` object as described in the previous section.

Consider *forward dynamic analysis* as implemented in the class `DynamicSolver`. In forward dynamic analysis we solve the equations

$$-f = 0, \tag{7a}$$
$$\tilde{g} = 0, \tag{7b}$$

for the acceleration v and λ, where f and $\tilde{v}$ are affine mappings of v and λ which have the form

$$-f = Mv + G_v^T\lambda - f', \tag{8a}$$
$$\tilde{g} = G_v\dot{v} - \gamma. \tag{8b}$$

Taking into account Equation (8), Equation (7) actually has the form of a linear constrained minimization problem and `DynamicSolver` is derived from a general solver for this kind of problem, passing it v, λ, $-f$, $\tilde{g}$, M and G_v.

To apply `DynamicSolver` to *any* of the models of the previous sections we declare the solver as

```
DynamicSolver dynamic_solver (system);
```

As explained in Section 4, the solution vector, consisting of v and λ, now behaves as a function of any other arguments of f and $\tilde{g}$, e.g. the position and velocity variables p and v. Thus, to compute accelerations for various states, in the simplest case we could use code like

```
for (...)
  {
    system.q_p = . ; // Set the state
    cout << system.qd_v, // Print the accelerations
  }
```

Other multibody solver classes implemented in this style are `KinematicSolver` for kinematic analysis, `StaticSolver` for static equilibrium and kinetostatic analysis, and `ODEIntegrator` or `DAEIntegrator` for numerical integration as described in [Anantharaman and Hiller, 1991].

An important feature of these multibody solver classes is that under certain circumstances they can be combined. Thus, suppose we wish to do forward dynamic analysis of the four-bar with joint-coordinates, but wish to perform a coordinate partitioning in which the joint-coordinates of the P_2 and U_3 are *numerically* expressed in terms of the joint coordinate of R_1. For this purpose we first assemble an auxiliary multibody system containing P_2, U_3 and the constraint for S_4 and apply a `KinematicSolver` to it as follows

```
System aux_system;
aux_system.add (P_2);
aux_system.add (U_3);
aux_system.add (S_4);

KinematicSolver kinematic_solver (aux_system);
```

As a result, the joint-coordinates of P_2 and U_3 now for all purposes behave as functions of the joint-coordinate of R_1 and the constraint for S_4 is identically satisfied. Therefore, we formulate the final multibody system from the rigid body and R_1 *only* and apply a `DynamicSolver` to it as follows

```
System system;
system.add (body);
system.add (R_1);

DynamicSolver dynamic_solver (system);

for (...)
  {
    system.q_p = . ; // Set the state
    cout << system.qd_v; // Print the independent accelerations
    cout << aux_system.qd_v; // Print the dependent accelerations
  }
```

Effectively, we have obtained a system with minimal coordinates using the same components from which we previously assembled a system with redundant coordinates

8 Conclusion

An attempt has been made to combine various existing methods for the analysis of multibody systems within an object-oriented programming environment. The resulting C++ toolkit can be used to quickly generate simulation programs and to experiment with various types of coordinates and methodologies. It can also be easily extended with new components or new analysis procedures.

Abstractions of the various components of multibody systems are implemented within the toolkit as C++ classes. With these classes it is possible to model a multibody system in a simple and intuitive manner: frames, links and joints are declared to describe the kinematical structure, constraints can be used to restrict motion and dynamic elements are introduced to represent applied forces by their virtual work. A multibody system is assembled from such components and various analysis procedures can be applied to it with just a few program statements. By adopting different strategies of modelling the kinematics and using different types of coordinates, one can apply either the absolute-coordinate, joint-coordinate, or minimal-coordinate approach to the same multibody system.

An important feature of the implementation is the combination of symbolical and numerical computations in a set of classes for mathematical operations. The symbolic operations greatly simplify the formulation of the higher-level classes for components of the multibody system, and allow operations on the complete multibody system to be performed in the same way for all types of coordinates. Thus generalized forces are formulated by symbolic differentiation of the virtual work with respect to virtual displacements and symbolic differentiation is also used to automatically formulate the system matrices in sparse form. On the other hand, by using techniques such as expression simplification, common subexpression elimination and value-caching one automatically obtains numerical efficiency comparable to that achieved by specialized algorithms used otherwise.

References

[Anantharaman and Hiller, 1991] Martin Anantharaman and Manfred Hiller. Numerical simulation of mechanical systems using methods for differential-algebraic equations. *Int. J. Num. Meth. Eng.*, 32:1531–1542, 1991.

[Bae and Haug, 1987] Dae-Sung Bae and Edward J. Haug. A recursive formulation for constrained mechanical system dynamics: Part I. Open loop systems. *Mech Struct. Mach* , 15(3) 359–382, 1987.

[Booch, 1991] Grady Booch. *Object Oriented Design with Applications.* Benjamin Cummings, 1991

[Cardona and Géradin 1991] Alberto Cardona and Michel Géradin. Modeling of superelements in mechanism analysis. *Int. J. Num. Meth. Eng.*, 32 1565–1593, 1991.

[Featherstone, 1987] Roy Featherstone. *Robot Dynamics Algorithms.* Kluwer Academic Publishers, Boston, 1987.

[Hiller and Kecskeméthy, 1989] M. Hiller and A. Kecskeméthy. Equations of motion of complex multibody systems using kinematical differentials. *Transactions of the CSME*, 13(4):113–121, 1989.

[Kecskeméthy, 1993] Andrés Kecskeméthy. *Objektorientierte Modellierung der Dynamik von Mehrkörpersystemen mit Hilfe von Übertragungselementen.* PhD thesis, Universität Duisburg, 1993.

[Nikravesh, 1987] P.E. Nikravesh. *Computer Aided Analysis of Mechanical Systems* Prentice-Hall, Englewood Cliffs, NJ, 1987

[Simo, 1985] J.C. Simo A finite strain beam formulation. The three-dimensional dynamic problem. Part I. *Comp. Meth. Appl. Mech. Eng.*, 49:55–70, 1985.

[Sorge, 1992] Kai Sorge *Mehrkörpersysteme mit starr-elastischen Subsystemen.* PhD thesis, Technische Universität München, 1992.

[Stroustrup, 1988] Bjarne Stroustrup. What is object-oriented programming. *IEEE Software Magazine*, pages 10–20, May 1988.

[Stroustrup, 1991] Bjarne Stroustrup. *The C++ Programming Language.* Addison-Wesley, second edition, 1991.

[Wittenburg, 1977] J. Wittenburg. *Dynamics of Systems of Rigid Bodies.* Teubner, Stuttgart, 1977.

[Woernle, 1988] C. Woernle. *Ein systematisches Verfahren zur Aufstellung der geometrischen Schließbedingungen in kinematischen Schleifen mit Anwendung bei der Rückwärtstransformation für Industrieroboter* PhD thesis, Universität Stuttgart, 1988.

NON-RECURSIVE METHODS FOR THE SOLUTION OF THE INVERSE DYNAMICS OF FLEXIBLE MULTIBODY SYSTEMS

R. LEDESMA and E. BAYO
Department of Mechanical Engineering
University of California
Santa Barbara, California 93106
U.S.A

ABSTRACT. This paper addresses the problem of end-point trajectory tracking in flexible multibody systems through the use of inverse dynamics. A global Lagrangian approach is employed in formulating the system equations of motion, and an iterative procedure is proposed to achieve end-point trajectory tracking in three-dimensional, flexible multibody systems. Each iteration involves firstly, a recursive inverse kinematics procedure wherein elastic displacements are determined in terms of the rigid body coordinates and Lagrange multipliers, secondly, an explicit computation of the inverse dynamic joint actuation, and thirdly, a non-recursive forward dynamic analysis wherein generalized coordinates and Lagrange multipliers are determined in terms of the joint actuation and desired end-point coordinates. In contrast with the recursive methods previously proposed, this new method is the most general since it is suitable for both open-chain and closed-chain configurations of three-dimensional multibody systems. The algorithm yields stable, *non-causal* actuating joint torques and associated Lagrange multipliers that account for the constraint forces between flexible multibody components.

1. Introduction

The effect of elastic deformation on the dynamics of multibody systems has been vigorously studied during the past thirty years. In particular, the modeling of multibody components as elastic beams has received considerable attention as made evident in the survey papers of Lowen and Jandrasits,[1] Erdman and Sandor,[2] Modi,[3] and more recently by Lowen and Chassapis.[4] A specific area of interest with regards to flexible multibody systems, especially in the aerospace and robotics industries, is in controlling the motion of a specified point in the multibody system. In most cases, the control objective is to have the end-point of the multibody system follow a desired trajectory. Various feedback control strategies for the problem of end-point trajectory tracking have been proposed, and the survey papers of Balas[5] and Book[6] present some of the approaches advanced by the controls community towards this problem.

The problem of end-point trajectory tracking in flexible multibody systems has led to the development of computational methods commonly referred to as inverse dynamics. Inverse dynamics deals with the problem of determining the joint actuation that will cause a specified control point in the flexible multibody system to follow a desired trajectory. The pioneering work of Reference 7 on the trajectory control of a single flexible link through inverse dynamics showed

that the inverse dynamic torque is *non-causal* with respect to the end-point motion, *i e* , actuation is required before the end-point has started to move as well as after the end-point has stopped. Moulin[8] demonstrated that because of the non-minimum phase character of the inverse dynamics for the trajectory tracking problem, the only bounded solution for the inverse dynamic torque has to be non-causal. Bayo, *et. al* .[9] extended the inverse dynamics to planar, multiple-link systems using an iterative frequency domain approach. The recursive method proposed in that study is suitable for planar open-chain systems, but required an *ad hoc* procedure for planar closed-chain systems. A time domain inverse dynamics technique based on the non-causal impulse response function was presented by Bayo and Moulin[10] for the single link system, with provisions for extension to multiple link systems. An equivalent time domain approach for a single link arm was proposed by Kwon and Book[11] where the non-causality of the computed torque was captured by dividing the inverse system into causal and anticausal parts. Recently, a more systematic and more general non-recursive, frequency domain method for the inverse dynamics of *planar* multibody systems was proposed in Reference 12. This method includes the constraint forces between the multibody components in the equations of motion, and the method is found suitable for both open-chain and closed-chain configurations of planar multibody systems. The effect of Coriolis forces and centrifugal forces on the inverse dynamics of constrained mechanical systems was presented by Gofron and Shabana.[13]

The inverse dynamics approach to end-point trajectory tracking of open-chain flexible multibody systems was recently applied to the three-dimensional problem by Ledesma, *et. al* ,[14] where a recursive procedure was proposed to simultaneously track a desired end-point trajectory and minimize motion-induced vibrations through the combined use of lumped inverse dynamic torques and distributed piezoelectric actuators. The recursive procedure required a controlled motor at each intermediate revolute joint and three motors at the ground. This procedure is effective for open-chain systems, but it is not valid for closed-chain systems because in such systems, the number of required control inputs is less than the number of joints.

In this paper, we present a general computational approach for the solution of the *non-causal* inverse dynamics of three-dimensional, flexible multibody systems, that is suitable for both open-chain and closed-chain configurations. With this work, we present a methodology that is suitable for all multibody systems, ranging from the single link case to three-dimensional systems with general topologies. The equations of motion are formulated in Section 2 and an iterative algorithm is subsequently developed. Simulation results for open-chain and closed-chain configurations are presented in Section 3 to demonstrate the validity and accuracy of the method.

2. Problem Formulation

In this section, we derive the governing equations of motion for a flexible multibody system by using a global Lagrangian approach, and develop a solution for the inverse dynamics problem that is suitable for both open-chain and closed-chain configurations. Crucial to the success of the proposed procedure is the use of the correct, *non-causal* Lagrange multipliers that account for the constraint forces between flexible multibody components. These Lagrange multipliers are determined in a forward dynamic analysis, which in turn, require the unknown inverse dynamic actuations. Therefore, the proposed solution is an iterative procedure which converges to the stable, non-causal inverse dynamic actuations and the associated Lagrange multipliers.

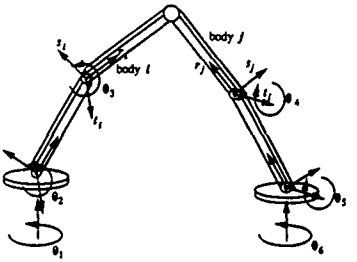


Fig. 1: A three-dimensional flexible multibody system

Consider an n-body flexible multibody system such as that shown in Fig. 1. A typical multibody component, say body i, is shown in Fig. 1 along with the floating reference frame associated with that body. The generalized coordinates consist of rigid body coordinates q_r^i which describe the position and orientation of the floating reference frame associated with each multibody component, and deformation coordinates q_f^i which describe the deformation of the flexible body with respect to its floating reference frame. The rigid body coordinates q_r^i consist of the Cartesian coordinates R^i which describe the position of the origin of the floating reference frame associated with body i, and a set of Euler parameters θ^i which describe the orientation of the floating frame. The use of Euler parameters among several choices of orientation coordinates will be explained later in the section describing the inverse dynamics solution procedure. The deformation from the nominal configuration is assumed to be small, so that the different bending and torsional modes are decoupled. For the sake of completeness, we summarize in the following equations the basic kinematic expressions that lead to the general equations of motion for flexible multibody systems. A more detailed formulation is found in Reference 15. With the above choice of coordinates, the position of an arbitrary point P in body i is given by

$$r^i = R^i + A^i u^i \tag{1}$$

where R^i is the location of the origin of the body axes with respect to the inertial frame, u^i is the location of point P with respect to the body axes, and A^i is the rotation transformation matrix from the body axes to the inertial frame. In the three-dimensional case, the rotation transformation matrix is given by

$$A^i = \begin{bmatrix} 2(\theta_0^2+\theta_1^2)-1 & 2(\theta_1\theta_2-\theta_0\theta_3) & 2(\theta_1\theta_3+\theta_0\theta_2) \\ 2(\theta_1\theta_2+\theta_0\theta_3) & 2(\theta_0^2+\theta_2^2)-1 & 2(\theta_2\theta_3-\theta_0\theta_1) \\ 2(\theta_1\theta_3-\theta_0\theta_2) & 2(\theta_2\theta_3+\theta_0\theta_1) & 2(\theta_0^2+\theta_3^2)-1 \end{bmatrix}^i \tag{2}$$

where the orientation coordinates are represented by four Euler parameters θ_0, θ_1, θ_2, and θ_3 which satisfy the following identity:

$$\sum_{k=0}^{3} (\theta_k^i)^2 = 1 \tag{3}$$

The position vector with respect to the body axes, u^i, can be decomposed into

$$u^i = u_r^i + u_f^i \tag{4}$$

where u_r^i is the position vector of point P in the undeformed state with respect to the body axes, and u_f^i is the deformation vector of point P with respect to the body axes. The deformation vector u_f^i can be expressed in terms of the nodal deformations by using a finite element discretization scheme, hence

$$u_f^i = N^i q_f^i \tag{5}$$

where N^i is the shape function matrix and q_f^i is the nodal deformation vector. Differentiating Eq. (1) with respect to time, we obtain the following expression for the velocity vector in terms of the rigid body coordinates and nodal deformation coordinates: [15]

$$\dot{r}^i = \dot{R}^i - 2\,A^i\,\tilde{u}^i\,E^i\,\dot{\theta}^i + A^i\,N^i\,\dot{q}_f^i \tag{6}$$

where $(\dot{})$ represents differentiation with respect to time, E^i is a matrix that depends linearly on the Euler parameters and is given by

$$E^i = \begin{bmatrix} -\theta_1 & \theta_0 & \theta_3 & -\theta_2 \\ -\theta_2 & -\theta_3 & \theta_0 & \theta_1 \\ -\theta_3 & \theta_2 & -\theta_1 & \theta_0 \end{bmatrix}^i \tag{7}$$

and $\tilde{u}^i$ is a 3 x 3 skew-symmetric matrix given by

$$\tilde{u}^i = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}^i \tag{8}$$

in which u_x, u_y, and u_z are the coordinates of the generic point P with respect to the body axes, in the deformed configuration.

Considering the reference coordinates $q^T = [\,R^T, \theta^T, q_f^T\,]$ as generalized coordinates for the flexible multibody system, these coordinates are not independent because the motion of specific points in different bodies are related according to the type of mechanical joints that interconnect them. Moreover, in flexible mechanical systems, the deformation of a component affects the configuration of adjacent components. As a consequence, the interdependence of the generalized coordinates is expressed by a vector of kinematic constraint equations, such as

$$\Phi(q,t) = 0 \tag{9}$$

where q is the total vector of system generalized coordinates, t is time, and Φ is the vector of linearly independent holonomic constraint equations. These constraint equations can be further classified into:

1 rigid body constraints where only rigid body variables are involved in the constraint equation,

2 joint constraints where both rigid body and deformation coordinates are included in the constraint equation; and

3 rheonomic constraints wherein the constraint equations can be explicit functions of time as well as generalized coordinates

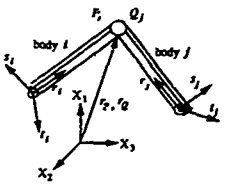


Fig. 2: A pair of flexible bodies connected by a spherical joint

The third type of constraint becomes active, for example, in the case of imposing the coordinates of the end-effector to follow a desired trajectory. To illustrate the construction of constraint equations, take the case of a spherical joint which connects two flexible bodies i and j at points P and Q shown in Fig. 2. The three constraint equations corresponding to the constraint condition that requires points P and Q to be coincident can be written as

$$\left(\mathbf{R}^i + \mathbf{A}^i \mathbf{u}_P^i\right) - \left(\mathbf{R}^j + \mathbf{A}^j \mathbf{u}_Q^j\right) = 0. \tag{10}$$

We note that the constraint equation exemplified by Eq. (10) forms a set of coupled nonlinear algebraic equations in the rigid body coordinates and deformation coordinates.

Considering the rigid body and deformation coordinates described above as generalized coordinates, and following standard procedures in multibody dynamics, the constrained equations of motion become [15]

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{C}\,\dot{\mathbf{q}} + \mathbf{K}\,\mathbf{q} + \boldsymbol{\Phi}_{\mathbf{q}}^{T}\,\boldsymbol{\lambda} = \mathbf{Q}_e + \mathbf{Q}_v(\mathbf{q},\dot{\mathbf{q}}) \tag{11}$$

where M, C and K are the system mass, damping and stiffness matrices, respectively, λ is the vector of Lagrange multipliers associated with the constraints, Φ_q is the constraint Jacobian matrix, Q_e is the vector of applied external forces, and Q_v is the quadratic velocity vector. The quadratic velocity vector contains the centrifugal forces and Coriolis forces that result from the differentiation of the kinetic energy expression with respect to the generalized coordinates. Geometric stiffening due to high rotation rates can also be added to the vector Q_v.

2.1 FORWARD DYNAMICS

In a forward dynamic analysis, *i e*, finding the resulting motion given the applied joint forces and external forces, Eqs (9) and (11) constitute a mixed system of differential-algebraic equations that have to be integrated simultaneously. As explained in the next section, the solution to the inverse dynamics problem requires a forward dynamic analysis within an iteration process. We

solve the forward dynamics problem by using the augmented Lagrangian penalty formulation [16] Applying the augmented Lagrangian penalty formulation to Eqs. (9) and (11) results in the following equation:

$$M(q)\,\ddot{q} + C\,\dot{q} + K\,q + \Phi_q^T\,\alpha\left[\ddot{\Phi} + 2\,\mu\,\omega\,\dot{\Phi} + \omega^2\,\Phi\right] = Q_e + Q_v(\dot{q},q) - \Phi_q^T\,\lambda^* \qquad (12)$$

where α is a diagonal matrix of penalty factors whose elements are large real numbers that will assure the satisfaction of constraints, ω and μ are diagonal matrices representing the natural frequencies and damping characteristics of the *dynamic penalty system* associated with the constraints. Values of α in the range $10^3 < \alpha < 10^6$ provide excellent results when working in double precision. The augmented Lagrangian method requires an iteration for the correct value of the Lagrange multipliers. The iterative equation for the Lagrange multipliers is given by

$$\lambda_{i+1}^* = \lambda_i^* + \alpha\left[\ddot{\Phi} + 2\,\mu\,\omega\,\dot{\Phi} + \omega^2\,\Phi\right] \qquad (13)$$

The iterative process described by Eq (13) involves only a few additional operations during each iteration but it significantly improves the convergence of the forward dynamics solution as compared to the standard penalty method.[16]

The augmented Lagrangian penalty formulation has several advantages over the standard algorithms used in solving differential-algebraic equations First, the method obviates the need to solve a mixed set of differential-algebraic equations and does not increase the number of equations to account for the constraints. Second, this method allows the use of standard unconditionally stable algorithms without the need of further stabilization techniques to control the violation of constraints during the integration process Third, the method can handle redundant constraints and allows the multibody system to undergo singular positions. Fourth, the constraint forces (Lagrange multipliers) can be obtained as a by-product of the integration without having to integrate additional equations for them. Finally, the method assures convergence independent of the penalty values used.

2.2 INVERSE KINEMATICS AND INVERSE DYNAMICS

Moulin and Bayo [17] showed that because of the non-minimum phase character of the inverse problem, the *unique stable solution* is found to be *non-causal, i e*, actuation is required before the end-point has started to move as well as after the end-point has stopped. These findings have been corroborated by Paden and Chen[18] in their theoretical work on the inversion of nonlinear non-minimum phase systems such as flexible multibodies. In addition, the fact that the stable solution starts at negative time and extends into future time precludes standard time domain integration schemes currently available in multibody computer codes from obtaining the proper inverse solution. These codes will yield causal results and therefore are valid, in general, for the forward dynamics only. The integration is of crucial importance in obtaining non-causal solutions, and as previously demonstrated in the planar inverse dynamics problem, the time-anticipatory effect can be automatically obtained by integrating in the frequency domain[9] or in the time domain by using the non-causal impulse response function and the bilateral Laplace transform.[10]

A previously proposed solution to the three-dimensional inverse dynamics problem[14] relied on a pinned-free finite element model of a flexible beam, and the equation for the inverse dynamics torque was formulated by imposing the condition that the torsional deformation and the two transverse deformations of the free end of each link be zero throughout the motion Thus limited type of model led to a recursive scheme to solve the inverse dynamics of flexible multibody

systems, and is found suitable for open-chain configurations but not for closed-chain configurations. In this section, we describe an iterative Lagrangian procedure to solve the three-dimensional inverse dynamics problem for either open-chain or closed-chain topologies. Our overall strategy is to first solve the inverse kinematics problem, *i.e.*, finding the unknown rigid body coordinates q_r and flexible body displacements q_f, given the desired end-point coordinates as explicit functions of time. Having determined the correct generalized coordinates and their time derivatives, the inverse dynamics joint torques can be obtained explicitly from the equations of motion. Compared to the recursive procedure cited above, this new approach is more systematic and becomes the only choice when closed-chain systems are encountered. We model the elastic links under pinned-pinned boundary conditions. Furthermore, since torsional deformations cause deviations from the nominal configuration further down the chain, we model the elastic link as fixed with respect to torsion at the distal end of the link.

Our goal then is to formulate an inverse kinematics equation that is linearized about the nominal motion, so that the elastic displacements, which are non-causal with respect to the end-point motion, can be determined through a frequency domain analysis. This is possible only if the leading matrix of the linearized equation is time-invariant and if the forcing term is Fourier transformable. This objective has been achieved in the planar case with the use of reference coordinates for the rigid body variables to describe the position and orientation of the floating reference frame.[12]

The three-dimensional inverse kinematics problem presents additional difficulties not found in the planar case. First, unlike the planar case, the three-dimensional torque vectors change directions in time, so that the external force vector Q_e in Eq (11) becomes a nonlinear function of the rigid body orientation coordinates. To overcome this difficulty, a proper parametrization of the rigid body coordinates and proper bases for the joint torques are necessary to attain the stated objectives in forming the linearized inverse kinematics equations. As described later in this section, the desired form of the linearized inverse kinematics equation is possible if Euler parameters are used to describe the rigid body orientation and if the base torque vector of each multibody component is expressed in terms of components along the associated floating reference frame.

A second difficulty that appears in the three-dimensional inverse dynamics problem is that the end-point vibration in the plane defined by the revolute joint axis and the member axis can not be controlled by the torque applied at the revolute joint. This suggests that additional actuation is necessary to control the end-point motion when the multibody system reaches an "inaccessible" configuration.[19] This problem has been addressed in Reference 14 wherein one motor at each intermediate joint and three motors at the ground were proposed to control the end-point motion for all possible configurations of a certain class of open-chain, flexible multibody systems. The problem of "inaccessibility" in open-chain systems, however, can be completely avoided simply through a judicious design of the orientation of the joint motors so that end-point vibration is controllable for all possible configurations. For closed-chain systems, "inaccessible" configurations do not occur, hence the controllability of the end-point motion is assured without the need for extra actuation.

Consider again the system equations of motion expressed by Eq (11). For a typical multibody component, say body i, the equations of motion can be written in the following partitioned form:

$$\begin{bmatrix} m_{RR} & m_{R\theta} & m_{Rf} \\ m_{\theta R} & m_{\theta\theta} & m_{\theta f} \\ m_{fR} & m_{f\theta} & m_{ff} \end{bmatrix} \begin{bmatrix} \ddot{R} \\ \ddot{\theta} \\ \ddot{q}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c_{ff} \end{bmatrix} \begin{bmatrix} \dot{R} \\ \dot{\theta} \\ \dot{q}_f \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_{ff} \end{bmatrix} \begin{bmatrix} R \\ \theta \\ q_f \end{bmatrix}$$

$$+ \begin{bmatrix} \Phi_R^T \\ \Phi_\theta^T \\ \Phi_f^T \end{bmatrix} \lambda = \begin{bmatrix} Q_{eR} \\ Q_{e\theta} \\ Q_{ef} \end{bmatrix} + \begin{bmatrix} Q_{vR} \\ Q_{v\theta} \\ Q_{vf} \end{bmatrix} \tag{14}$$

The elements of the mass matrix and quadratic velocity force vector corresponding to an isoparametric, three-dimensional curved beam finite element are given in the Appendix.

Let us define τ^i as the torque vector at the base of body i, whose three components τ_1^i, τ_2^i, and τ_3^i are parallel to the associated floating reference axes r^i, s^i, and t^i, respectively. If we use Euler parameters as the rigid body orientation coordinates, the externally applied joint forces $Q_{e\theta}^i$ associated with the rigid body rotation of body i can be expressed as

$$Q_{e\theta}^i = [G^i]^T \{\tau^i - [A^i]^T A^{i+1} \tau^{i+1}\} \tag{15}$$

where τ^i is the base torque acting on body i and whose components are parallel to the floating reference axes associated with body i; τ^{i+1} is the vector of joint torques and reaction moments transmitted from body i to body $i+1$, and whose components are parallel to the floating reference axes associated with body $i+1$; A^i and A^{i+1} are body axes to inertial axes rotation transformation matrices for bodies i and $i+1$, respectively; and G^i is a matrix that maps the derivatives of the Euler parameters describing the orientation of the reference frame of body i to the angular velocity of this reference frame, and is given by $G^i = 2\,E^i$. Combining Eq. (15) with the second set of equations in Eq. (14) yields

$$[G^i]^T \{\tau^i - [A^i]^T A^{i+1} \tau^{i+1}\} = m_{\theta R}^i \ddot{R}^i + m_{\theta\theta}^i \ddot{\theta}^i + m_{\theta f}^i \ddot{q}_f^i + \Phi_\theta^T \lambda - Q_{v\theta}^i \tag{16}$$

If we pre-multiply both sides of Eq. (16) by $\frac{1}{4}G^i$ and use the identity

$$\frac{1}{4} G\, G^T = I_3 \tag{17}$$

where I_3 is the 3×3 identity matrix, we can get the following result after expanding the inertia matrices and quadratic velocity vectors found in Eq. (16), (the reader is referred to the Appendix for the expressions for the inertia matrices and quadratic velocity vectors in terms of the invariants of the motion):

$$\tau^i = [A^i]^T A^{i+1} \tau^{i+1} + \frac{1}{4} G^i\, m_{\theta R}^i \ddot{R}^i + J^i\, G^i \ddot{\theta}^i + J_f^i \ddot{q}_f^i + \frac{1}{4} G^i \Phi_\theta^T \lambda$$

$$+ \frac{1}{2} G^i \{\dot{G}^i\}^T \{J^i\, G^i \dot{\theta}^i + J_f^i \dot{q}_f^i\} \tag{18}$$

where J^i is the 3×3 inertia tensor of body i with respect to the origin of the floating reference frame and measured relative to this frame, and J_f^i is the inertia matrix coupling the rigid body rotation and the elastic deformation. The key to obtaining a time-invariant leading matrix, that is necessary in transforming the linearized equations of motion into the frequency domain, is the fact that the inertial coupling matrix J_f^i can be decomposed into the sum of a time-invariant matrix and a time-varying matrix, *i.e.*,

$$J_j = J_{jc} + J_{jt} \tag{19}$$

where J_{jc} and J_{jt} are the time-invariant part and time-varying part of J_j, respectively. This decomposition is essential to the formulation of the inverse kinematics equations that lead to non-causal solutions to the nonlinear inversion problem. This is also the reason for selecting Euler parameters as rigid body orientation coordinates over other types of singularity-free coordinates such as natural coordinates,[20] where the decomposition of the inertial coupling matrix into time-invariant and time-varying parts is multiplicative rather than additive as in Eq. (19). Introducing this decomposition into Eq. (18) results in the following expression for the base torque on body i:

$$\tau^i = [A^i]^T A^{i+1} \tau^{i+1} + T_{bj} + J_{jc}^i \ddot{q}_j \tag{20}$$

where T_{bj} is a torque vector given by

$$T_{bj} = \frac{1}{4} G^i m_{dR}^i \ddot{R}^i + J^i G^i \ddot{\theta}^i + J_{jt}^i \ddot{q}_j^i + \frac{1}{4} G^i \Phi_{\theta}^T \lambda$$
$$+ \frac{1}{2} G^i [\dot{G}^i]^T (J^i G^i \dot{\theta}^i + J_j^i \dot{q}_j) \tag{21}$$

Considering the equations of motion associated with the elastic degrees of freedom, the externally applied force vector due to the joint torques acting on body i can be expressed as

$$Q_{tf}^i = N_b^T \tau^i - N_t^T [A^i]^T A^{i+1} \tau^{i+1} \tag{22}$$

where N_b and N_t are the shape function matrices associated with a torque vector acting on node b (base) and at node t (tip) of the finite element mesh, respectively. Combining Eq. (22) with the third set of equations in Eq. (14) yields the following inverse kinematics equations for body i:

$$\bar{m}_{jf}^i \ddot{q}_j^i + c_{jf}^i \dot{q}_j^i + k_{jf}^i q_j^i = F^i(\lambda, q_j^i, \dot{q}_j^i, \ddot{q}_j^i, q_j^i, \dot{q}_j^i, \ddot{q}_j^i) \tag{23}$$

where the modified mass matrix is given by

$$\bar{m}_{jf}^i = m_{jf}^i - N_b^T J_j^i \tag{24}$$

and the motion-induced force vector acting on the elastic degrees of freedom is given by

$$F^i = N_b^T ([A^i]^T A^{i+1} \tau^{i+1} + T_{bj}) - N_t^T [A^i]^T A^{i+1} \tau^{i+1}$$
$$+ Q_{jf}^i - \Phi_{qj}^T \lambda - m_{jR}^i \ddot{R}^i - m_{j\theta}^i \ddot{\theta}^i \tag{25}$$

The modified mass matrix $\bar{m}_{jf}^i$ is nonsymmetric and it is precisely this nonsymmetry that produces elastic displacements which are non-causal with respect to the end-point motion when non-causal techniques are employed to obtain the proper inversion of the nonlinear, non-minimum phase systems. Furthermore, inspection of Eqs.(23)-(25) shows that the inverse kinematics equation for body i assumes that the base torque vector τ^{i+1} is known beforehand. This suggests some form of recursive algorithm for the inverse kinematics, *i.e.*, finding the elastic displacements starting from the end-point, and proceeding to the base of the multibody system (inboard direction). This procedure is straightforward for open-chain configurations. However, for closed-chain configurations, we need to take the additional step of cutting the chain at the joint that is defined as the end-point, and then proceed as in the open-chain case, since the constraint forces at the cut are automatically accounted for by the vector of Lagrange multipliers.

The nonlinear inversion can now be carried out efficiently in the frequency domain since the leading matrices have been constructed such that they remain constant throughout the motion. Our strategy is to solve Eq. (23) in the frequency domain to obtain the nodal deformation vector

q_f that is non-causal with respect to the end-point motion. In the frequency domain, Eq (23) can be written as a set of complex equations for a particular frequency ω

$$\left[\bar{m}_{ff} + \frac{1}{i\omega} c_{ff} - \frac{1}{\omega^2} k_{ff}\right] \hat{\ddot{q}}^f(\omega) = \hat{F}^f(\omega) \tag{26}$$

where $\hat{\ddot{q}}^f(\omega)$ is the Fourier transform of $\ddot{q}^f(t)$ and $\hat{F}^f(\omega)$ is the Fourier transform of $F^f(t)$. Eq (26) is based on the assumption that $\ddot{q}^f(t)$ and $F^f(t)$ are Fourier transformable This assumption is valid for slewing motions which are from rest to rest. The nodal acceleration vector $\hat{\ddot{q}}^f(\omega)$ can be obtained directly from Eq. (26) for each frequency ω. The leading matrix of Eq (26) is a complex regular matrix that is invertible for all frequencies except for $\omega = 0$ However, for $\omega = 0$, the system undergoes a rigid body motion, and the leading matrix will be determined only by m_{ff} which is positive definite and therefore invertible. We note, however, that the forcing vector on the right hand side of Eq. (26) depends on the elastic deformations, velocities and accelerations Therefore, an iterative process is needed to obtain the solution to the differential equations which are nonlinear in q^f. We start the iteration process by assuming zero elastic deformations, velocities and accelerations for the initial calculation of the forcing vector $F^f(t)$, and use a successive substitution scheme to converge to the correct solution. Finally, the elastic displacements and their derivatives in the time domain may be obtained through the application of the inverse Fourier transform, *e g.*,

$$\ddot{q}^f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\ddot{q}}^f(\omega)\, e^{i\omega t}\, d\omega \tag{27}$$

Alternately, the computation of the elastic displacements and their derivatives in each iteration can also be carried out in the time domain through the use of the non-causal impulse response function and the bilateral Laplace transform, *e g.*,

$$\ddot{q}_f(t) = \int_{-\infty}^{\infty} \sum_{j=1}^{j=m} h_j(t-\tau)\, f_j(\tau)\, d\tau \tag{28}$$

where $h_j(t)$ is the non-causal acceleration response vector to an impulse applied to the j^{th} degree of freedom and $f_j(t)$ is the j^{th} component of the forcing term on the right hand side of Eq. (23) We note that the integration from $-\infty$ to ∞ is necessary to capture the non-causal effects.

Once the non-causal elastic displacements and their derivatives are known, Eq. (18) can be used to explicitly compute the non-causal inverse dynamics joint efforts that will move the end effector according to a desired trajectory. We note, however, that the joint torques and elastic displacements given by Eqs. (18) and (23), respectively, depend on the Lagrange multipliers and rigid body coordinates, which in turn depend on the elastic displacements and the applied torque. Moreover, the rigid body coordinates and Lagrange multipliers are different from their nominal values when the components of the multibody system are flexible. Therefore, a forward dynamic analysis is required to obtain an improved estimate of the generalized coordinates and Lagrange multipliers. In order to ensure that the iteration process converges to obtain the joint efforts that will cause the end-effector to follow the desired trajectory, the forward dynamics analysis is carried out with the additional constraint that the coordinates of the end-point follow the desired trajectory. These additional constraints have corresponding Lagrange multipliers which act as correcting terms to the joint efforts that have been previously calculated

To summarize, the procedure for obtaining the inverse dynamics solution for three-dimensional, flexible multibody systems involve the following steps:

Algorithm:

1. Perform a rigid body inverse dynamic analysis to obtain the nominal values of the rigid body coordinates q_r and Lagrange multipliers λ.
2. Solve the inverse kinematics equation in the frequency domain through Eq. (23) or in the time domain through Eq. (28) to obtain the time-delayed elastic displacements and their time derivatives
3. Compute the inverse dynamics joint efforts τ using Eq. (18)
4. Perform a forward dynamic analysis using Eqs. (12) and (13), to obtain new values for the generalized coordinates and Lagrange multipliers.
5. Repeat steps 2 through 4 until convergence in the inverse dynamics torques is achieved.

It is worthwhile to compare the recursive procedure proposed in Reference 14 and the algorithm proposed in this paper. The most important difference between the two methods is that the former method assumes that the dependence of rigid body coordinates on the elastic displacements are made negligible through the action of control forces so that the rigid body coordinates take on values corresponding to the nominal motion. This assumption is not made in the present method and consequently, the solution of the inverse kinematics equation of Eq. (23) would require an iteration for the rigid body coordinates q_r as well as the Lagrange multipliers λ that are needed as inputs to the inverse kinematics equation. A consequence of the above assumption in the previously proposed recursive procedure is that control inputs were required at all intermediate joints in the multibody system. This requirement is acceptable in open-chain configurations, but not practical in closed-chain configurations because the number of system degrees of freedom is less than the number of joints in a closed-chain multibody system. The present procedure takes advantage of this fact and allows the analyst to choose *a priori* which joints in the multibody system are the control joints. Therefore, the present algorithm is more general and more systematic than the previously proposed procedure, although it requires more computational effort.

3. Simulation Results and Discussion

We present in this section some results of the numerical implementation of the procedure discussed above. First, we apply the procedure proposed in this paper to the inverse dynamics of a two-link, open-chain flexible multibody system undergoing motion in three-dimensional space, and compare the results with those obtained by the previously proposed recursive procedure.[14] Next, we present some simulation results of the application of the present procedure to the inverse dynamics of a closed-chain, flexible multibody system undergoing three-dimensional motion.

3.1 OPEN-CHAIN MULTIBODY SYSTEM

The iterative procedure discussed in the preceding section is applied to the three-dimensional open-chain flexible manipulator shown in Fig. 3. The multibody system is controlled by three motors at the base and one motor at the intermediate revolute joint. The desired motion is to have the end-point remain in the x_2–x_3 plane with the x_2 coordinate and x_3 coordinate of the end-point following the trajectories shown in Fig. 4. Gravitational forces are neglected. The two links share

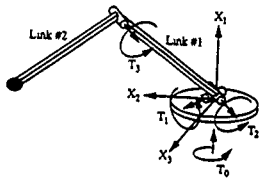


Fig. 3: Two-link, flexible open-chain multibody system.

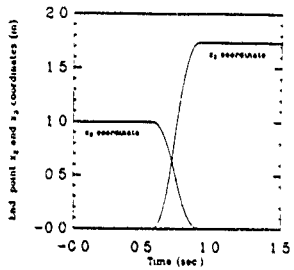


Fig. 4: Nominal end-point coordinates of the open-chain system.

the following geometric and material properties:

Length: 1 0 *m*
Cross section dimensions: 1.0 *cm* x 1 0 *cm*
Young's modulus: 70 *GPa*
Shear modulus: 27 *GPa*
Mass density: 2715 kg/m^3
Tip mass: 0 1 *kg*

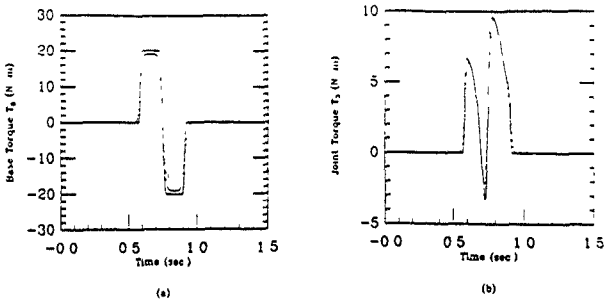


Fig 5: Open-chain inverse dynamic torques: (a) joint torque T_0; (b) joint torque T_3

We perform two sets of computations for the example considered: 1) using the recursive procedure proposed in Reference 14; and 2) using the iterative procedure proposed in this paper. In the open-chain case, each intermediate joint needs to be controlled, and we therefore expect very similar results from both methods. Plots of inverse dynamic joint torques needed to track the desired end-point trajectory are shown in Figs. 5a and 5b. The results obtained from the two methods superimpose, thus validating the method proposed in this paper. Plots of the corresponding rigid body torques are also shown in the figures to illustrate the non-causal nature of the inverse dynamic torques. In Figs. 5a-b, the dashed curves refer to the inverse dynamic torques, while the solid curves refer to the rigid body torques.

3.2 CLOSED-CHAIN MULTIBODY SYSTEM

Fig. 6 shows a closed-chain, three-dimensional flexible multibody system, where the selected control torques are shown in the figure. Joints 1-4 are revolute joints while joint 5 is a spherical joint. The desired end-point (joint 5) trajectory is a motion in the x_2–x_3 plane with the x_2 coordinate and x_3 coordinate of the end-point following the trajectories shown in Fig. 7. As in the open-chain case, gravitational forces are not considered in order to focus on the inertial effects on the dynamics of the system. The four links share the following geometric and material properties:

Length: 1.0 *m*
Cross section dimensions: 1.0 *cm* x 1.0 *cm*
Young's modulus: 40 *GPa*
Shear modulus: 15 *GPa*
Mass density: 2715 kg/m^3
Tip mass: 0.1 *kg*

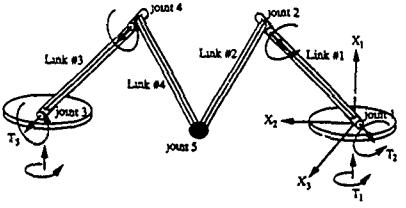


Fig. 6: Three-dimensional, flexible closed-chain system.

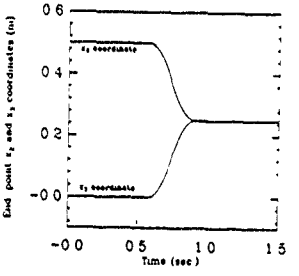


Fig. 7: Nominal end-point coordinates of the closed-chain system

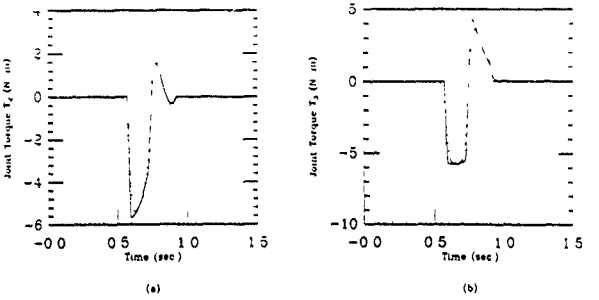


Fig 8: Closed-chain inverse dynamic torques: (a) joint torque T_2, (b) joint torque T_3

The present procedure is applied to the closed-chain system by introducing a cut at the end-point (joint 5), thus creating two open-chain systems. The internal constraint forces exposed by the cut are automatically taken into account by the Lagrange multipliers in the equations of motion. Figs. 8a and 8b show joint torques T_2 and T_3, respectively, that are needed to achieve the desired end-point trajectory. In these figures, the dashed curves refer to the inverse dynamic torques obtained by the present procedure, while the solid curves refer to the corresponding rigid body torques. We observe that pre-actuation and post-actuation with respect to end-point motion are exhibited by the inverse dynamic torques.

4. Conclusion

We have presented a new iterative procedure for determining the inverse dynamic torques that are needed for end-point trajectory tracking in three-dimensional flexible multibody systems. An iterative procedure is necessary because of the interdependence between the elastic coordinates, the rigid body coordinates and the associated Lagrange multipliers in the system equations of motion. This procedure is valid for both open-chain and closed-chain configurations, and differs from the previously proposed recursive procedures in the sense that the rigid body coordinates are not assumed to follow the nominal motion. The conditions for trajectory tracking are now met in a more general way through the satisfaction of rheonomic constraint conditions. The new method is shown to yield the same results as those obtained with the recursive procedures for open-chain systems with normal link flexibilities and normal rotation rates. For closed-chain systems, however, this new method is the only valid procedure for determining the inverse dynamic torques since in this case, the number of control torques is smaller than the number of joints and therefore, the recursive methods can not be applied.

Further research is needed to address the problem of simultaneous trajectory tracking and vibration minimization of three-dimensional, flexible multibody systems with general topologies In a future paper, we will address this problem through the use of redundant actuation in the form of combined lumped and distributed actuators.

Acknowledgement

The support of this work by the Air Force Office of Scientific Research under contract no F49620-91-C-0095 is gratefully acknowledged.

Appendix

The elements of the mass matrix and quadratic velocity force vector in Eq (14) can be expressed in terms of the so-called invariants of the motion which need to be computed only once at the start of the simulation. For each component of the flexible multibody system, the invariants of the motion can be expressed by the following integrals:

$$Z_1 = \int_V u_r \, \rho \, dV \tag{A 1}$$

$$Z_2 = \int_V N \rho \, dV \tag{A 2}$$

$$Z_3 = \int_V \begin{bmatrix} (u_{r_2}^2 + u_{r_3}^2) & -u_{r_1}u_{r_2} & -u_{r_1}u_{r_3} \\ -u_{r_2}u_{r_1} & (u_{r_3}^2 + u_{r_1}^2) & -u_{r_2}u_{r_3} \\ -u_{r_3}u_{r_1} & -u_{r_3}u_{r_2} & (u_{r_1}^2 + u_{r_2}^2) \end{bmatrix} \rho \, dV \tag{A.3}$$

$$Z_{ij} = \int_V u_{r_i} N_j \, \rho \, dV, \quad i, j = 1,2,3 \tag{A 4}$$

$$Z_{ij} = \int_V N_i^T N_j \, \rho \, dV, \quad i, j = 1,2,3 \tag{A 5}$$

where ρ is the mass density, V is the volume of the component, N is the shape function matrix, and N_j is the j^{th} row of the shape function matrix. We observe that the motion invariant Z_1 is a measure of the first moment of the undeformed component about the body axes, and the motion invariant Z_3 is the inertia tensor of the undeformed component with respect to the body axes

Closed-form expressions for the motion invariants corresponding to the three-dimensional, Bernoulli-Euler straight beam element are given in Reference 15. In this paper, however, we use the variable-node, isoparametric, three-dimensional curved beam element developed by Bathe and Bolourchi[21] to model the flexible links As a result, the motion invariants can be expressed in terms of integrals which are evaluated numerically through Gaussian quadrature

The components of the mass matrix, expressed in terms of the invariants of the motion are given by the following

$$m_{RR} = m \, I_3 \tag{A 6}$$

$$m_{R\theta} = -A \, \hat{S} \, G \tag{A 7}$$

$$m_{Rf} = A\, Z_2 \tag{A 8}$$

$$m_{\theta\theta} = G^T\, J\, G\,, \quad J = [Z_3 + J_1 + J_2] \tag{A 9}$$

$$m_{\theta f} = G^T\, J_f\,; \quad J_f = [J_{fc} + J_{ft}] \tag{A 10}$$

$$m_{ff} = Z_5^{11} + Z_5^{22} + Z_5^{33} \tag{A 11}$$

where, in Eq (A 6), m is the total mass of the component, and the tilde symbol above the vector in Eq (A.7) refers to the skew-symmetric matrix operator The matrices S, J_{fc}, J_{ft}, J_1, and J_2 are given by

$$S = Z_1 + Z_2\, q_f \tag{A 12}$$

$$J_{fc} = \begin{bmatrix} Z_4^{23} - Z_4^{32} \\ Z_4^{31} - Z_4^{13} \\ Z_4^{12} - Z_4^{21} \end{bmatrix} \tag{A 13}$$

$$J_{ft} = \begin{bmatrix} q_f^T (Z_5^{23} - Z_5^{32}) \\ q_f^T (Z_5^{31} - Z_5^{13}) \\ q_f^T (Z_5^{12} - Z_5^{21}) \end{bmatrix} \tag{A 14}$$

$$J_1 = \begin{bmatrix} (p_{22} + p_{33}) & -p_{12} & -p_{13} \\ -p_{21} & (p_{11} + p_{33}) & -p_{23} \\ -p_{31} & -p_{32} & (p_{11} + p_{22}) \end{bmatrix} \tag{A 15}$$

$$J_2 = \begin{bmatrix} (q_{22} + q_{33}) & -q_{12} & -q_{13} \\ -q_{21} & (q_{11} + q_{33}) & -q_{23} \\ -q_{31} & -q_{32} & (q_{11} + q_{22}) \end{bmatrix} \tag{A 16}$$

in which

$$p_{ij} = [Z_4^{ij} + Z_4^{ji}]\, q_f\,, \quad i, j = 1,2,3 \tag{A 17}$$

and

$$q_{ij} = q_f^T\, Z_5^{ij}\, q_f\,, \quad i, j = 1,2,3 \tag{A 18}$$

The quadratic velocity force vectors are given by

$$Q_{vR} = -A\, \{\tilde{\omega}^2\, S + 2\, \tilde{\omega}\, Z_2\, \dot{q}_f\} \tag{A 19}$$

$$Q_{v\theta} = -2\, \tilde{G}^T\, \{J\, \omega + J_f\, \dot{q}_f\} \tag{A 20}$$

$$Q_{vf} = -[\tilde{\omega}^2]^*\, [Z_4^{11} + Z_4^{22} + Z_4^{33}]^T - [\tilde{\omega}^2]^*\, m_{ff}\, q_f - [2\, \dot{\tilde{\omega}}]^*\, m_{ff}\, q_f \tag{A 21}$$

where ω is the absolute angular velocity of the body axes, whose components are expressed with respect to the body axes and given by

$$\omega = G\, \dot{\theta} \tag{A 22}$$

and the matrices $[\tilde{\omega}^2]^*$ and $[2\,\tilde{\omega}]^*$ are block diagonal matrices whose diagonal elements are $\tilde{\omega}^2$ and $2\tilde{\omega}$, respectively.

References

1 Lowen, G. G. and Jandrasits, W. G., "Survey of Investigations into the Dynamic Behavior of Mechanisms Containing Links with Distributed Mass and Elasticity," *Mechanism and Machine Theory*, vol. 7, pp. 3-17, 1972.

2. Erdman, A. G. , A Sandor, G. N., "Kineto-elastodynamics - a Review of the State of the Art and Trends," *Mechanism and Machine Theory*, vol. 7, pp. 19-33, 1972.

3 Modi, V. J., "Attitude Dynamics of Satellites with Flexible Appendages - a Brief Review," *Journal of Spacecraft and Rockets*, vol. 11, pp. 743-751, 1974.

4 Lowen, G. G. and Chassapis, C., "The Elastic Behavior of Linkage: An Update," *Mechanism and Machine Theory*, vol. 21, pp. 33-42, 1986.

5. Balas, M. J. , "Trends in Large Space Structure Control theory: Fondest Hopes, Wildest Dreams," *IEEE Transactions on Automatic Control*, vol. AC-27, no. 3, pp. 522-535, 1982.

6 Book, W. J. , "Modeling, Design, and Control of Flexible Manipulator Arms: Status and Trends," *Proceedings, NASA Conference on Space Telerobotics*, vol. 3, pp 11-24, January, 1989.

7. Bayo, E., "A Finite-Element Approach to Control the End-Point Motion of a Single-Link Flexible Robot," *Journal of Robotic Systems*, vol. 4, no. 1, pp. 63-75, 1987.

8 Moulin, H , "Problems in the Inverse Dynamics Solution for Flexible Manipulators," *PhD Thesis*, University of California, Santa Barbara, 1989.

9 Bayo, E., Papadopoulos, P., Stubbe, J , and Serna, M., "Inverse Kinematics and Dynamics of a Multi-Link Elastic Robot: An Iterative Frequency Domain," *International Journal of Robotics Research*, vol. 8, no. 6, pp. 49-62, 1989.

10 Bayo, E., and Moulin, H., "An Efficient Computation of the Inverse Dynamics of Flexible Manipulators in the Time Domain," *Proceedings, IEEE Conference on Robotics and Automation*, pp 710-715, 1989

11 Kwon, D. S and Book, W J., "An Inverse Dynamic Method Yielding Flexible Manipulator State Trajectories," *Proceedings, American Control Conference* , San Diego, California, 1990

12 Ledesma, R. and Bayo, E , "A Non-Recursive Lagrangian Approach to the Inverse Dynamics of Flexible Multibody Systems: The Planar Case," *International Journal for Numerical Methods in Engineering*, 1993 To appear

13 Cyfron, M. and Shabana, A., "Effect of the Deformation in the Inertia Forces on the Inverse Dynamics of Flexible Mechanical Systems," *Nonlinear Dynamics*, 1993 To appear

14 Ledesma, R., Devasia, S., and Bayo, E , "Simultaneous Trajectory Tracking and Vibration Reduction of Distributed Parameter Systems through Inverse Dynamics," *Proceedings, 14th ASME Biennial Conference on Mechanical Vibration and Noise*, Albuquerque, New Mexico, September, 1993

15 Shabana, A A , *Dynamics of Multibody Systems*, John Wiley & Sons, Inc , 1989

16. Bayo, E., Garcia de Jalon, J., Avello, A. and Cuadrado, J., "An Efficient Computational Method for Real Time Multibody Dynamic Simulation in Fully Cartesian Coordinates," *Computer Methods in Applied Mechanics and Engineering*, vol. 92, pp. 377-395, 1991.

17. Moulin, H., and Bayo, E. , "On the End-point Trajectory Tracking for Flexible Manipulators through Non-causal Inverse Dynamics," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 113, no. 2, pp. 320-324, June 1991.

18. Paden, B. and Chen, D., "Zero Dynamics and Inversion of Nonlinear Systems," *Proceedings, U.S.-Japan Conference on Flexible Automation*, San Francisco, California, 1992.

19. Tosunoglu, S., Lin, S., and Tesar, D., "Accessibility and Controllability of Flexible Robotic Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 114, pp. 50-58, March 1992.

20. Garcia de Jalon, J., Unda, J., and Avello, A., "Natural Coordinates for the Computer Analysis of Multibody Systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 56, pp. 309-327, 1986.

21. Bathe, K. J. and Bolourchi, S., "Large Deformation Analysis of Three-Dimensional Beam Structures," *International Journal for Numerical Methods in Engineering*, vol. 14, no. 7, pp. 961-986, 1979.

A MULTI LEVEL APPROACH TO SYNTHESIS OF PLANAR MECHANISMS

M. R. Hansen, Ph D
Assistant professor
Institute of Mechanical Engineering
Aalborg University
Pontoppidanstræde 101
9220 Aalborg Ø

ABSTRACT A multilevel approach to synthesis of planar mechanisms is presented. The approach covers both structural and dimensional synthesis of planar rigid body mechanisms containing revolute and translational joints The synthesis is based on four different criteria. Firstly the type of mechanism is chosen with a view to get the simplest mechanism that satisfactorily fulfill the remaining three criteria Two of these criteria are formulated as constraints on the kinematic behavior and the total area occupied by the mechanism, respectively The fourth criteria is simply the desired minimization of the reactive forces/moments that appear in the mechanism The desired kinematic behavior is based on a finite number, typically 1..6, of points in time (positions of the mechanism) where the position and orientation of up to two output bodies may be prescribed. The constraints on occupied areas are labelled territory constraints and formulated as a number of restricted areas (boxes). A synthesis is automatically performed at five levels At the first level the structure of the mechanism is decided At the second level initial dimensions for the given type of mechanism are found by random checking At the third level the constraints on the kinematic behavior is fulfilled. At the fourth level the territory constraints are taken into account and, finally, at the fifth level the minimization of reactions is carried out The entire approach has been implemented in a software package SYNMEC that runs on PC's and constitutes a way of performing the synthesis of a mechanism, that is general and flexible with respect to both the type of mechanism that may be synthesized as well as the desired behavior upon which the synthesis is based.

1. Introduction

Whenever motion or power is to be transmitted from one point to another the designer faces a number of decisions that must be taken in order to get on with the design process:

- mechanical/not mechanical transmission
- gear-/cam-/linkage-/combined mechanism
- number and type of drivers
- structure

The evaluation of these decisions can only be done by applying some kind of dimensional synthesis taking into account the different demands and wishes of the specifife problem Synthesis

of mechanisms typically involves a number of criteria of varying nature and complexity There are however, three types of criteria that almost always can be identified in practical examples

- kinematic behavior
- surroundings
- dynamic behavior

Since the original work by Fox and Willmert [2] on mechanism synthesis using minimization techniques much attention has been paid to problems whith both kinematic and dynamic behavior as criteria, e g. [1] and [5]. In the current work a procedure for synthesis of mechanisms is presented that take into account all three types of criteria Also some of the decision making is covered. The approach is inspired by the work by Garrett and Hall [3] that used random search as part of the synthesis procedure and the work by Soni et. al [6] where the mechanism type is decided on the basis of the desired behavior of the output body

The presented approach should be thought of as a tool that, on the basis of prescribed constraints on the kinematic behavior and the surroundings, returns a number of simple planar mechanisms with optimized dynamic behavior that act as support to the designer in deciding whether to use mechanisms and if so which one to pick. The types of mechanisms that are taken into account have one independent input and only contain lower pair joints. Hence, prior to using the procedure, the designer must choose to spend time investigating whether mechanical transmission in the shape of planar linkage mechanisms with these characteristics is a useful solution to the problem at hand.

The basic idea of the presented procedure is the multi level approach. The input is the constraints on the kinematic behavior (prescribed motion for one or two output bodies) and the territory constraints (restricted areas) A total of five levels are recognized. At the first level the structure of the mechanism is decided based on the number of output bodies and the type of motion prescribed. At the second level initial dimensions for the given type of mechanism are found by random checking. This level is also only based on the prescribed kinematic behavior At the third level the constraints on the kinematic behavior is fulfilled exactly by minimization At the fourth level the territory constraints are fulfilled and, finally, at the fifth level the minimization of reactions (optimization of dynamic behavior) is carried out.

The entire method has been implemented in a computer program SYNMEC and frequently in the remaining paper references will be made to the default settings in SYNMEC that influence the automated perfomance of the synthesis

2. Synthesis input

In order to perform any synthesis it is necessary with some input representing the task to be carried out. In the current work the desired kinematic behavior as well as the territory constraints (restricted areas) are required.

The kinematic behavior is the basis of any mechanism synthesis. It corresponds to the motion of the output body(ies) of the mechanism and it is always prescribed in some way, often as a function of the motion of the input body(ies) Classically this prescribed/desired kinematic behavior is discretized into a relatively limited number (here 1 6) of points in time where positions and/or rotations of output and/or input bodies are of significance to the functionality of the mechanism These points in time will be referred to as **positions of synthesis**. The main

advantage of looking at the mechanism in a limited number of points is the reduced time of computation. The main disadvantage is that the mechanism might act in an unexpected way between the positions of synthesis

In the current implementation up to two output bodies may be defined as either rotating, translating, or general motion, see Figure 2 1

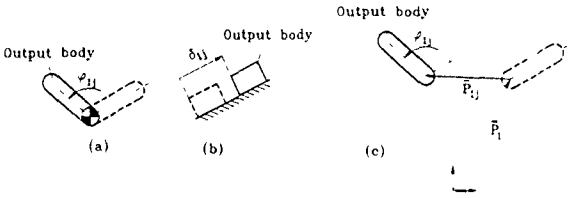


Figure 2.1 (a) rotating output body, (b) translating output body, (c) general motion output body Each body is shown in the j'th position of synthesis (continuous line) and the 1st position of synthesis (dotted line). The different quantities that are prescribed are shown

For a rotating output body the rotation relative to the first position of synthesis is prescribed

$$\phi_{1j} \quad , \quad j = 2 \quad n_{pos}$$

where n_{pos} is the number of positions of synthesis. For a translating output body the translation relative to the first position of synthesis is prescribed

$$\delta_{1j} \quad , \quad j = 2 \quad n_{pos}$$

For a general motion output body there exist a point of interest. The rotation of the body and the motion of the point of interest relative to the first position of synthesis are prescribed In order to relate the point of interest relative to the surroundings its position in the first position of synthesis is also prescribed:

$$\bar{P}_{1j} = (P_{1jx}, P_{1jy}) \quad , \quad j = 2 \quad n_{pos}$$

$$\phi_{1j} \quad , \quad j = 2 \quad n_{pos}$$

$$\bar{P}_1 = (P_{1x}, P_{1y})$$

Each variable may be prescribed in three different ways

1 X is free
2 X = C
3 $C_1 < X < C_2$

where X denotes any variable and the C's are the prescribed constants

Restricted areas are simply defined as a number of boxes, see Figure 2 2. The mechanism must stay outside these boxes in any position (not only the positions of synthesis) The advantage of using boxes of restriction is that it is possible to generate both convex, and concave outer limits as well as isolated areas, e g , a beam passing thorugh the plane in which the mechanism is to be synthesized

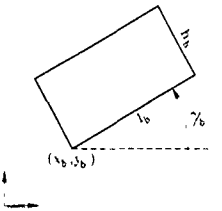


Figure 2 2. Box of restriction area. The box is given by five parameters. x_b, y_b, γ_b, l_b, and h_b

3. Mechanism structure

There is no such thing as an optimal mechanism structure for a given task There are, however, some simple rules such as the fewer movable bodies the better and rather revolute joints than translational joints According to this it is always a good idea to try and solve the problem using a four bar mechanism as the one shown in Figure 3.1.

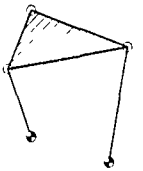


Figure 3 1 Four bar mechanism

For one reason or another the four-bar mechanism might not be a good choice Naturally, if the desired kinematic behavior is prescribed as two general motion output bodies it is necessary to look at a mechanism with more movable bodies that are not directly connected to the frame Normally, however, the choice of structure can only be evaluated after a number of dimensional syntheses has been performed The results could indicate that it is impossible to fulfill all constraints or that the dynamic behavior lies at an unacceptable level subject to these constraints Then it is necessary to investigate another and more complex structure or simply to realize that it is not a good idea to use one degree of freedom mechanisms to the current task

The philosophy used in this work is based on the above observations and it says depending on the number of output bodies and the prescribed motion for these bodies a list of mechanism types is generated The first type is the simplest and, thus, the one that is checked first and so on Each check involves the remaining four levels described in the previous section carried out a specific number of times The designer may stop as soon as an acceptable design is reached or as soon as it becomes clear that this mechanism type cannot solve the problem satisfactorily
In Figure 3 2 the generated lists for some of the different possible output combinations are shown

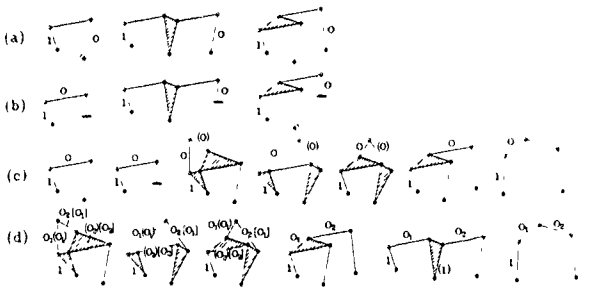


Figure 3.2. Mechanism types listed in that sequence they are checked for different combinations of output bodies (a) one rotating output body, (b) one translating output body, (c) one general motion output body, and (d) two general motion output bodies

In Figure 3 2 The letters i and o (with indices added in case of two output bodies) stands for input- and output body, respectively Alternative output bodies for the same structure are shown as (2nd alternative) and [3rd alternative]. The only exception to the restricted use of lower pair joints is the double crank (no. 7 in 3rd list) where it is understood that some kind of gear prescribes the rotation of the two bodies connected to the frame relative to each other
Naturally the designer may rearrange the sequence of each list or totally omit one or more types In the current implementation a fixed total of 18 mechanism types with up to five movable bodies are previously defined. If the designer wants to check more complicated types he must also define the structure in the structure editing facilities of SYNMEC Each mechanism is stored as Assur Groups and cranks (see also [4]) which makes it possible to have a both general and effective set of analysis routines for the calculation of positions, velocities, accelerations, and reactive forces and moments Also the subdivision in Assur Groups is of great advantage in the general identification of design variables (see Section 4), calculation of analytical sensitivities (see Section 5), and line segments (see Section 6)

4. Initial search

After the selection of a mechanism type the next step is to add some dimensions The simplest, and in the authors view, most effective way of doing this is by generating a large number of mechanisms and picking out a specific number that best fulfill the desired kinematic behavior

It could be argued that also the territory constraints should be included in this random search. This is omitted, however, because it is much more time consuming to check if a mechanism has moved into restricted areas in any position rather than checking its kinematic behavior in a few positions of synthesis Thus, the initial search returns a number of mechanisms with a kinematic behavior relatively close to the desired but with no regard to the surroundings.

In order to randomly generate dimensions for a mechanism type, two things a necessary identification of design variables and upper and lower limits for each of these The design variables may be divided into two categories geometrical and kinematical variables The geometrical variables make up the actual dimensions of the mechanism Consider the four-bar mechanism in Figure 4.1

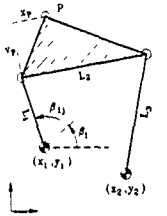


Figure 4.1 Four-bar mechanism shown in the j'th position of synthesis The coupler link is used as a general motion output body

For this mechanism the nine geometrical variables are:

$$x_1,\ y_1,\ x_2,\ y_2,\ L_1,\ L_2,\ L_3,\ x_P,\ y_P$$

The kinematical variables are the relative rotations of the input body in each position of synthesis as well as the rotation in the first position of synthesis:

$$\beta_1 \qquad \beta_{1j} \ ,\ j = 2,\ n_{pos}$$

Hence, the design variables $\bar{X}$ are made up by a fixed number of geometrical variables and a number of kinematical variables depending on the number of positions of synthesis

$$\bar{X} = (x_1,\ y_1,\ x_2,\ y_2,\ L_1,\ L_2,\ L_3,\ x_P,\ y_P,\ \beta_1,\ \beta_{12},\ \ldots,\ \beta_{1n_{pos}})$$

In general there are also one or more configurational parameters that describe how the mechanism is assembled. For the four-bar mechanism there is one such parameter These configurational parameters are part of the variables that decide the behavior of the mechanism, but they should not be mixed with the design variables because they always are fixed during a dimensional synthesis and therefore belong to the structural data In SYNMEC the different

configurations are taken into account during the initial search by simply checking each configuration for each randomly selected set of design variables

The automated identification of the design variables and their their upper and lower limits is done simultaneously Firstly, some characteristic length must be determined for the problem at hand. In the current work the desired kinematic behavior is used to generate this number by taking the average of the sizes of the prescribed translations/motions If only relative rotations are prescribed the characteristic length is simply set to 1 The limits are automatically prescribed as follows

$$\frac{L_c}{10} < \textit{length} < 3L_c$$

$$-3L_c < \textit{coordinate} < 3L_c$$

$$0 < \textit{rotation} < 2\pi$$

where L_c denotes the characteristic length The designvariables for the four bar mechanism in Fig 4 1 may be grouped in the following way

$$\textit{lengths} \quad L_1,\ L_2,\ L_3$$

$$\textit{coordinates} \quad x_1,\ y_1,\ x_2,\ y_2,\ x_P,\ y_P$$

$$\textit{rotations} \quad \beta_1,\ \beta_{12},\ \ldots,\ \beta_{1n_{pos}}$$

Two types of design variables, namely the position of revolute joints fixed to the frame as well as the relative rotations of the input body (timed problems), may be prescribed in the same manner as the desired kinematic behavior (see Section 2).

Two numbers are of practical interest in the initial search: the number of designs that are tested and the number of designs that are stored for further use. In the current version of SYNMEC the automatically selected values for these numbers are $20000 \cdot n_{config}$ and 20, respectively The number n_{config} represents the number of configurational parameters for the mechanism type in question It should be recognized that the purpose of the initial search not is to cover the entire design space (this is, even for simple mechanisms in a relatively f v positions of synthesis, a computationally discouraging task) but to get a number of reasonably good initial designs

5. Kinematic behavior

The initial search returns a number of designs that have a kinematic behavior close to the desired one Next step is to change these mechanisms with a view to fulfill the desired kinematic behavior exactly. Both during the initial search and the subsequent synthesis a formulation of the deviation from the desired kinematic behavior is required. For that purpose the function I is introduced

$$I(X) = \begin{cases} (X - C_1)^2 & , \ X < C_1 \\ 0 & , \ C_1 \le X \le C_2 \\ (X - C_2)^2 & , \ X > C_2 \end{cases}$$

Here C_1 and C_2 represents lower and upper limits, respectively, that are prescribed for the variable X If X is exactly prescribed to some value C this corresponds to

$$C_1 = C_2 = C$$

If X is a free variable we have

$$C_1 = -\infty \quad , \quad C_2 = \infty$$

In the case of a rotating output body the deviation is formulated as:

$$D_K = \sum_{j=2}^{r_{pos}} I\,(\phi_{1j})$$

The index K refers to kinematic deviation. In the case of a translating output body the deviation is formulated as:

$$D_K = \sum_{j=2}^{n_{pos}} I\,(\delta_{1j})$$

Finally, for a general motion body the deviation is formulated as:

$$D_K = I\,(P_{1x}) + I\,(P_{1y}) + \sum_{j=2}^{n_{pos}} I\,(P_{1jz}) + I\,(P_{1jy}) + w\,I\,(\phi_{1j})$$

A weightfactor w is introduced to eliminate the difference in magnitude between [length^2] and [angle^2]. In SYNMEC the automatically chosen value for w is:

$$w = 4L_c^2\ \frac{mm^2}{rad^2}$$

which approximately corresponds to saying that an angular deviation of 30° should equal a length deviation of L_c.

The minimization of D_K is performed using a Fletcher-Davidon-Powell routine The sensitivities are calculated analytically which increases the speed of calculation significantly.

As an example consider the four-bar mechanism in Figure 5.1 which is one of the results of an initial search

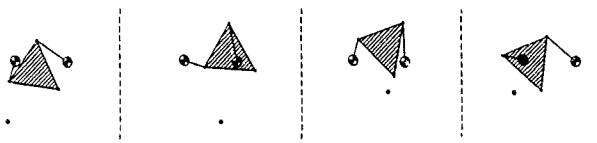


Figure 5 1 Initial mechanism shown in four positions of synthesis The desired position of the point of interest P in each positions of synthesis is indicated by a rectangle

The desired behavior is as follows

$$\overline{P}_1 = (0, 0) \quad \overline{P}_{12} = (75, 0) \quad \overline{P}_{13} = (75, 50) \quad \overline{P}_{14} = (0, 50)$$

Furthermore, the relative input rotations are prescribed to

$$80° < \beta_{12} < 100° \quad 170° < \beta_{13} < 190° \quad 260° < \beta_{14} < 280°$$

The minimization returns the mechanism shown in Figure 5 2.

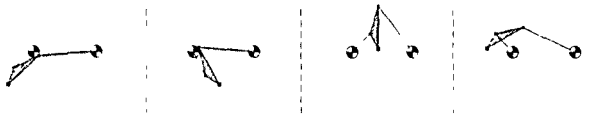


Figure 5 2. Synthesized mechanism. Kinematic constraints has been taken into account and the point of interest passes exactly through the four prescribed rectangles.

It should be noted that the mobility of the mechanism not is checked in the positions between the positions of synthesis.

6. Surroundings

Whether a mechanism is to be fitted into an existing environment or designed together with its environment there will always be some kind of requirements upon the amount of space that may be occupied. These territory restrictions can be of a very varying nature, e.g , some constraints are only active in certain periods (other moving machinery) and some only refers to some of the bodies of the mechanism

In order to examine the area swept by a mechanism it is necessary to analyze it in a number of positions that satisfactorily describes its continuous behavior. In SYNMEC a mechanism is by default analyzed in 35 positions between the first position of synthesis and the last.

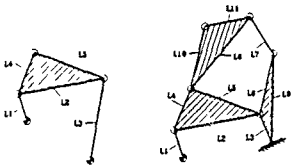


Figure 6 1 To mechanisms subdivided into five and eleven line segments, respectively.

As mentioned in Section 2 the territory constraints are defined as a number of fixed boxes In SYNMEC it is automatically assumed that all bodies should stay outside any box at any time The area occupied by a mechanism is best obtained if the mechanism is divided into line segments In Figure 6 1 the subdivision of two mechanisms into five and eleven line segments, respectively, is shown.

Each line segment must stay outside each box of restriction in each analyzed position. How much a line is inside (violates) a box is illustrated in Figure 6 2 where three out of four possible incidents that may occur between a line segment and a box are shown. The fourth incident corresponds to the segment lying outside the box in which case $\Delta = 0$

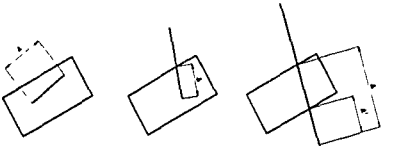


Figure 6 2. A box of restriction and a line segment that represents part of a mechanism are shown in three different incidents· (a) the line segment lies inside the box, (b) the line segment crosses the edges of the box once, and (c) the line segment crosses the edges of the box twice For each case the corresponding violation of the box Δ is shown.

In the case where the line segment crosses the edges twice it is necessary to include Δ' to avoid that the minimization routine just positions the line segment perpendicular to the crossed edges Denoting the violation of the i'th box of restriction by the j'th line segment Δ_{ij} it is possible to set up an expression for the total deviation from the territory constraints in a single analyzed position:

$$D_T = \frac{1}{n_B n_L} \sum_{i=1}^{i=n_B} \sum_{j=1}^{j=n_L} \Delta_{ij}^2$$

where the number of boxes of restriction and line segments are denoted n_B and n_L, respectively. The index T refers to territoty. The division by the total numbers of violations is done in order to get an average value that directly can be added to the kinematic deviation D_K without the risk of being dominating. In SYNMEC only the maximum value of D_T, as opposed to summing D_T from each analyzed position, is included in the minimization. This is, naturally, done because of the immense gain in computational efficiency The drawback is that a minimization step where only one analyzed position is considered might cause even higher values of D_T in other analyzed positions In SYNMEC this problem has been met by including in the minimization the three analyzed positions that most recently have contained the maximum D_T value.

A deviation that take into account both kinematic and territory constraints can now be stated as

$$D_{tot} = D_K + D_{T,max}$$

where $D_{T,max}$ is the sum of D_T values of the three previously mentioned positions. The minimization of D_{tot} can be performed directly on a mechanism delivered by the initial search

or to a mechanism where D_K already has been minimized The latter is the default approach in SYNMEC.

As an example the mechanism in Figure 5 2 is considered, see also Figure 6 3

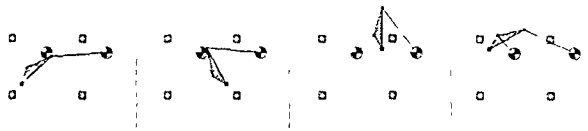


Figure 6 3 The mechanism in Figure 5 2 with territory constraints added. Four beams lying perpendicular to the plane has to be avoided by the mechanism The beams are indicated as hatched rectangles.

Four beams lying perpendicular to the plane are introduced as four boxes of restriction and a minimization of D_{tot} returns the mechanism shown in Figure 6 4

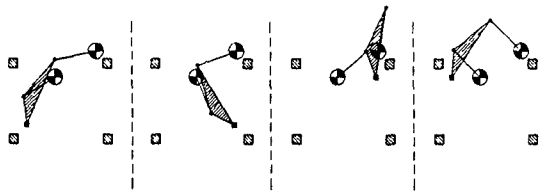


Figure 6 4 Synthesized mechanism Kinematic and territory constraints have been taken into account

It should be noted that SYNMEC by default only examines the period between the first and the last position of synthesis. It must be stated specifically if the territory constraints should be active for a full 360° rotation of the input body.

Because the mechanisms are thought of as lines the designer must take into account that the dimensions of the real mechanism takes up more space (width of links, bearings etc.) by defining tougher territory constraints.

7. Dynamic behavior

Any mechanism will be subject to some kind of applied loads as well as inertia forces. These loads cause reactive forces/moments to appear in the kinematic joints as well as the input forces/moments required to generate the desired motion In this context the dynamic behavior is recognized as those reactions Consequently the purpose of optimizing the dynamic behavior

corresponds to minimizing the sum of the squared reactive forces. If the entire dimensional synthesis is regarded as an optimization problem it should be formulated as:

Minimize reactive forces
subject to kinematic and territory constraints.

It is the experience of the author that the main problem in dimensional synthesis is to simply find a mechanism that meets both the constraints on the kinematic behavior as well as the surroundings. Hence, the difficult part is to get into the feasible design area whereas the subsequent improvement in dynamic behavior is less of a problem.

It is typical that a relatively modest change (at least to the human eye) of the design variables of a mechanism may give at significant improvement in the dynamic behavior.

No applied loads need to be defined. The output body(ies) are simply subjected to unit loads as shown in Figure 7.1.

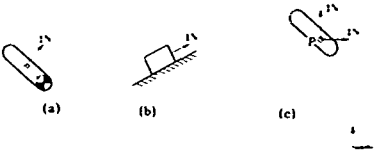


Figure 7.1 The automatically applied loads used on (a) a rotating output body, (b) a translating output body, and (c) a general motion output body.

The approach taken in SYNMEC is very stable but also rather slow. In order to avoid the problems of weighting fundamentally different quantities such as reactive forces and lengths/angles the calculation of the dynamic behavior is isolated from the calculation of the deviation from the constraints. This is done in the following way: a mechanism is generated that fulfils the constraints. The first design variable that is not prescribed to be exactly equal to some constant is perturbed a certain amount. This means that the mechanism no longer fulfils the kinematic and territory constraints and a minimization of D_{tot} is performed but with the perturbed variable fixed to its new value. If D_{tot} is reduced to an acceptable value, i.e., the value it had before the perturbation, then a new feasible mechanism has been generated and its dynamic behavior may be compared with that of the original one. This is repeated for each design variable and the pertubation that caused the greatest improvement in the dynamic behavior is selected as the new design. Now the entire proces of perturbing each design variable may be repeated and so on. The minimization of the reactive forces is repeated until none of the fixed pertubations caused any improvement or (which is what normally happens) D_{tot} cannot be minimized to an acceptable level for any of the dynamically improved designs. The size of the pertubations are

coordinates and lengths: $L_c/20$

rotations: 1°

It should be noted that a new characteristic length L_c is calculated as the average of all link lengths.

The method is basically quite slow because of the many minimizations of D_{tot} but a feature that strongly increases the efficiency has been introduced: the first variable that is perturbed in any except the first step is the one that gave the greatest improvement in the previous step. If this perturbation followed by the minimization D_{tot} still causes the dynamic behavior to be improved it is used again without checking the other perturbations. Naturally, one might risk to overlook a perturbation that gives a better improvement but in practise it works very well.

The dynamic behavior in a specific analyzed position is calculated as:

$$D_{dyn} = \sum_{i=1}^{n_R} R_i^2$$

where n_R is the number of reactions and R are the reactions. It should be noted that if R_i is a moment it is divided by the updated characteristic length L_c. The dynamic behavior used as the object of minimization is the maximum value of D_{dyn}. In order to find that one, the continuous behavior of the mechanism is required just as in the case of the territory constraints and, hence, the same number of analyzed positions are used.

As an example the mechanism in Figure 6.4 is subjected to a minimization of the reactions and the result is shown in Figure 7.2.

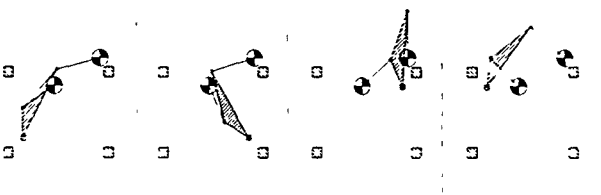


Figure 7.2 Synthesized mechanism. Kinematic and territory constraints as well as dynamic behavior has been taken into account

Although not much has happened the obtained improvement in $D_{dyn\,max}$ is 22%

8. Example

In this section an example that demonstrates the main idea of the approach discussed in this paper is given. The variety of solutions that may be obtained should give the designer a fair chance of either picking a reasonable design or choosing not to use the types of mechanisms handled in SYNMEC to solve the problem at hand.

A body is to be approximately vertically translated inside a square room, see Figure 8.1.

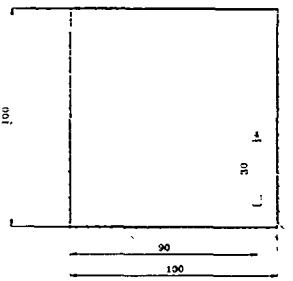


Figure 8 1. The kinematic and territory constraints The mechanism should stay within the square and the point of interest of the output body should move inside the vertical band to the right During the vertical motion the output body should not rotate more than 10° to either side

The kinematic behavior is accordingly prescribed as follows

$\overline{P}_1 = (90,10) \quad -2 < P_{12x} \; P_{14x} < 2 \quad P_{12y} = 10 \quad P_{13y} = 20 \quad P_{14y} = 30$

$-10° < \phi_{12} \; \phi_{14} < 10°$

Furthermore, the following constraints are laid upon the relative input rotations

$30° < \beta_{12} < 60° \quad 70° < \beta_{13} < 100° \quad 110° < \beta_{14} < 140°$

The territory constraints are defined by four large boxes of constraint that make sure the mechanisms do not move outside the room, see Figure 8 2

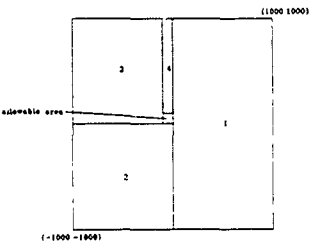


Figure 8 2 The four boxes of restriction that define the territory constraints

A small portion of the usable mechanisms are shown in Figure 8 3a e No dynamic behavior has been optimized for any of these mechanisms

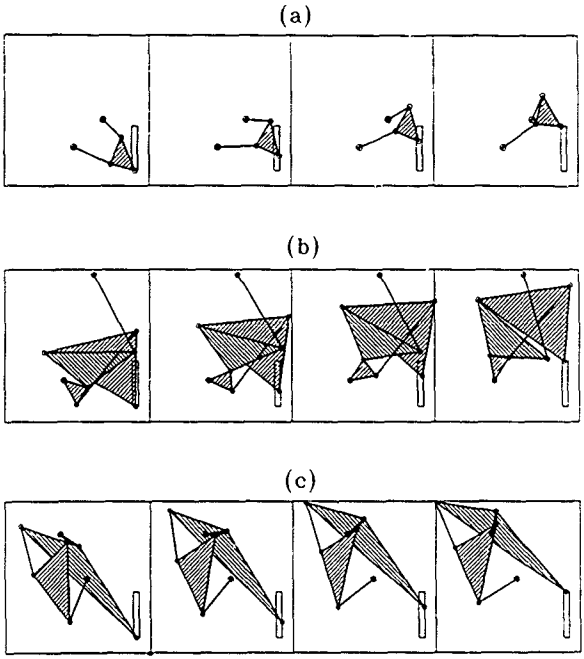
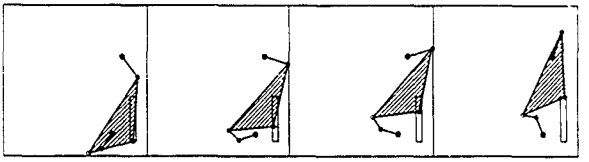


Figure 8 3a. Synthesized mechanisms, (a) four bar mechanism (b) six bar mechanism, (c) six bar mechanism (same structure but as (b) but alternative output body), continued on the next page, (d) another six bar mechanism, and (e) five bar mechanism (double crank)

(d)

(e)



9. Conclusion

In the current paper an approach to the combined structural and dimensional synthesis of planar rigid body mechanisms is introduced. The automated version of the approach is emphasized with a view to present a method that only requires input from the designer concerning the actual problem (and nothing concerning the mechanisms) in order to return a number of synthesized mechanisms Both kinematic and territory constraints may be prescribed in a new and general way An optimization of the dynamical behavior subject to these constraints may optionally be carried out.

The approach is ..ided into five levels that include picking a mechanism type, doing a random initial search, and performing dimensional synthesis with respect to kinematic behavior first, then territory constraints, and, finally, dynamic behavior

Future work will a o. concentrate on more complex ways of defining the territory constraints, e g , taking into account that some obstacles only are present at certain times and only have influence at certain bodies of the mechanism Also the possibility of defining any shape of particular bodies will be included

Acknowledgement. I would like to thank my colleagues John Hansen, Finn Uldum, and Carsten Nilsson at the Technical University of Copenhagen for advice and help

References

[1] Conte, F L , George, G R , Mayne, R W., and Sadler, J P , "*Optimum Mechanism Design Combining Kinematic and Dynamic-Force Considerations,*" Journal of Engineering for Industry, 1975

[2] Fox, R L , and Willmert, K D , "*Optimum Design of Curve-Generating Linkages With Inequality Constraints,*" Journal of Engineering for Industry, 1969

[3] Garrett, R.E , and Hall, A S , "*Optimal Synthesis of Randomly Generated Linkages,*" Journal of Engineering for Industry, 1968

[4] Hansen, M R., "*Development and Implementation of Methods for Analysis and Synthesis of Mechanical Mechanisms,*" Ph D.-thesis, Institute of Mechanical Engineering, Aalborg University, 1992

[5] Kakatsios, A.J., and Tricamo, S.J , "*Design of Planar Rigid Body Guidance Mechanisms With Simultaneously Optimized Kinematic and Dynamic Characteristics,*" Design Engineering Technical Conference, Ohio, 1986

[6] Soni, A H , Dado, M H F , and Weng, Y , "*An Automated Procedure for Intelligent Mechanism Selection and Dimensional Synthesis,*" Journal of Mechanisms, Transmissions, and Automation in Design, 1988

SYNTHESIS OF SPATIAL MECHANISMS USING OPTIMIZATION AND CONTINUATION METHODS.

JOHN M HANSEN
Department of Solid Mechanics
Building 404
Technical University of Denmark
DK-2800 Lyngby
DENMARK

ABSTRACT The aim of this work has been to develop a method that can be used within synthesis of spatial mechanisms, specifically on the problem of designing a mechanism for which a coupler point can describe a given path, given by a number of discrete points

In order to maintain the method general, i e be able to work for arbitrary mechanisms, a general analysis method is used as the basic tool, and to maintain a minimal number of degrees of freedom the joint, or relative, formulation is employed No attemps are being made to match the points exactly Instead an optimization procedure is employed to minimize the distance from the path of the coupler point and the desired path

In order to stabilize the procedure, a continuation method is applied in conjunction with the optimization method so that the desired path are gradually changed from points being close to points on the initial coupler curve to the ones actually desired With this combination a method for synthesizing a spatial mechanism towards a mechanism that can describe a given coupler curve has been developed that is both general and stable

1. Introduction

In this paper the problem of pathgeneration for spatial mechanisms is treated I e , a procedure is developed with which dimensions for a given type of mechanism may be determined in order for a point on the mechanism to traverse a given path.

For planar mechanisms this has been dealt with for many years and both textbooks and papers treating the subject are available, e.g. [1-9] The general approach in most presentations have been to focus on one specific type of mechanism and develop a method for synthesizing that particular mechanism Only few papers treat more general formulations that are capable of dealing with arbitrary mechanisms Examples of this are HAUG & SOHONI [3] and HANSEN [10]

Synthesis of spatial mechanisms has not been explored to the same extent yet Among the work published ANGELES [11] and CHIEN *et al.* [12] give interesting examples These papers, however, deal with specific mechanisms, and the methods are not applicable for arbitrary mechanisms

In this paper a synthetis method is introduced based on an analysis program that can

deal with arbitrary spatial mechanisms It is thus capable of synthesizing general spatial mechanism to generate a given path.

The analysis method is based on the joint variable formulation, described for closed loop dynamics of rigid bodies in NIKRAVESH & GIM [13], so that a minimal set of degrees of freedom is used during the analysis, making the analysis phase very efficient

In the synthesis phase, the distance between the desired spatial curve and the one actually generated by the current design of the mechanism is minimized. This is carried out by making a spline interpolation of both curves, and then measure the distance as the sum of squares of two sets of points equally spaced along each curve. The synthesis phase consists of a minimization of this distance by a standard optimization procedure, letting a number of the dimensions of the mechanism be the design variables

As already observed by others, e.g. [8], [11], the minimization procedure applied to mechanisms may not be able converge if the initial guess is not very close to the desired configuration This problem may be overcome by using a continuation method, [8], [10], [11] The basic principle behind this is to define a series of intermediate paths, that are defined by interpolation between the path generated by the initial design and the desired one. The minimization procedure is then carried out sequentially for each of the intermediate paths This stabilizes the procedure considerably.

2. Analysis method

The analysis method is based on the joint coordinate formulation as described in [13] This method constitutes an efficient analysis tool with which arbitrary mechanisms can be analysed. The concepts of the method is summarized here. For a more detailed explanation the reader is referred to [13] or NIKRAVESH [14]

Assume that the j'th joint in a mechanism is a revolute joint, that connects body $j-1$ and body j, as shown in Fig. 1.

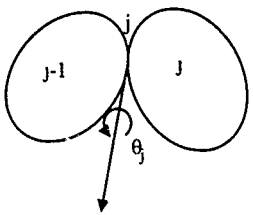


Fig 1 A revolute joint connecting two bodies The joint variable is θ_j

If the position, $\mathbf{r}_{j-1}$, and rotation, $\mathbf{A}_{j-1}$, as well as the revolute joint axis, $\mathbf{u}_j$, and the joint variable θ_j are known, the position $\mathbf{r}_j$ and rotation $\mathbf{A}_j$ of body j can be determined (Boldface symbols denote vectors and matrices).

Similar relations are valid for other types of joints, and for a chain of bodies, the position and rotation of each body may be found if the orientation of the base body and the vector θ of joint coordinates along the chain are known This is readily extended to more complex tree-structures, so that, knowing the position and rotation of the root body and the joint coordinates of all the joints in the tree, the orientation of all bodies in the tree may be determined.

Mechanisms, however, usually includes one or more closed loops. To deal with this, the loops are cut at one of the joints in each of the loops, and mathematical expressions are derived that expres the loop closure conditions For the revolute joint, e.g., there are three Cartesian coordinate conditions that ensure that the two bodies connected by the joint actually meet at the joint, and two alignment conditions that ensure that the only relative rotation between the two bodies is rotation about the revolute joint axis

The conditions are assembled in the constraint vector Ψ and the loop closure conditions are written as

$$\Psi(\theta) = 0 \tag{1}$$

To this vector the driver constraints are added in a manner similar to one found in e.g NIKRAVESH [15], but expressed in joint coordinates. The set of nonlinear equations are solved in a series of discrete positions of the drivers using the Newton-Raphson procedure as

$$\begin{aligned} \mathbf{C}_k \cdot \Delta\theta_{k+1} &= -\Psi(\theta_k) \\ \theta_{k+1} &= \theta_k + \Delta\theta_{k+1} \end{aligned} \tag{2}$$

Here, $\mathbf{C}$ is $\partial\Psi/\partial\theta$ and the expressions that appears in $\mathbf{C}$ are given in [14] for all standard combinations of joints. Eq (2) may then be solved with respect to the joint variables in a number of discrete time steps, corresponding to the desired positions of the drivers.

In Eqs (1) and (2) only the actual degrees of freedom in each joint enter as variables resulting in a very effective analysis tool

Once the joint variables are determined in each time step, the position and rotation of each body may be calculated as explained above The point of interest used to generate the desired path is on, say, body j Knowing the position and orientation of this body, the global position of the point of interest is calculated as

$$\mathbf{s}^P = \mathbf{r}_j + \mathbf{A}_j \mathbf{s}'^P \tag{3}$$

in which $\mathbf{s}'^P$ and $\mathbf{s}^P$ are the coordinates of the point of interest in the local coordinate system of body j, and in the global coordinate system, respectively

With this procedure any mechanism based on the joints for which expressions for the

updating process (from body j-1 → body j) and the entries in C have been derived, may be analyzed In the present work this includes the revolute, the translational and the spherical joint.

3. Path generation

The objective of this work is to be able to design a mechanism for which a coupler point traverses a given path. This is accomplished by minimizing the distance between the given path and the one actually generated by the mechanism. Hence, a measure of the distance is needed In this work a 3D-spline representation is employed, as will be shown below. During the minimization a "black-box" subroutine that solves the problem as a sequential quadratic programming problem is used.

3 1 THE 3D-SPLINE REPRESENTATION

Consider the two spatial curves shown in Fig 2 It is required to determine the distance between the desired curve, C_d , shown as a number of discrete points in Fig 2, and part of the curve, that the current mechanism generates, C_c For now, assume that it is known which point on the curve is the first and which is the last that is taken into consideration The curve has been determined numerically by the analysis procedure described earlier, and is thus given as a number of discrete points

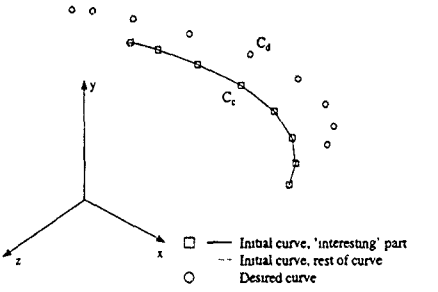


Fig 2 A desired path, given by discrete points, and the current path with the "interesting" part highlighted

The points that make up the two curves that are to be compared are not necessarily equal in number, and even if they are, there may not be a one-to-one correspondance between the two set points Hence, in order to compare them it is necessary to perform a

spline approximation for each of the curves, and from that generate a new set of points on each curve that can be compared.

The spline representation for each of the curves is found as follows As the curve is traversed from point to point, the (linear) distance between two adjacent points is calculated, see Fig. 3 The distances are summed and saved, so that an approximate curvelength for each curve is obtained Each partial length is then divided by the total curvelength, so that a series of discrete curveparameters u_i are found I e , the points are expressed as functions of u_i as

$$\begin{Bmatrix} x_n \\ y_n \\ z_n \end{Bmatrix} = \begin{Bmatrix} x(u_n) \\ y(u_n) \\ z(u_n) \end{Bmatrix} \quad , \quad n = 0, \ ,N \quad \text{and} \quad u_n = 0,....,1 \tag{4}$$

assuming there are N+1 points and N lines connecting them The curve parameter u is in the interval [0,1] Each of the set of coordinates, x , y , and z are then spline interpolated using a cubic spline routine, with u as spline parameter

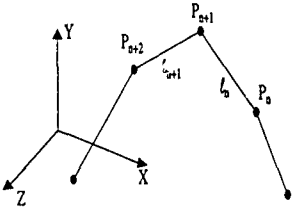


Fig 3 Points along a curve, and corresponding lenghts

3 2 THE OBJECTIVE FUNCTION

From the spline representation of the two curves it is now possible to generate a new series of points along each curve, using equidistant values of u from 0 to 1 . Thus two sets of points are generated that will converge towards the same points as the two curves approach each other.

Assuming M + 1 points are generated along each curve, a measure of the distance between the two curves may now be calculated as

$$\chi = \sum_{m=0}^{M} \left| P_m^d - P_m^c \right|_2 \tag{5}$$

in which χ is the difference between the curves, P_m^d is the m'th (spline interpolated)

point on the desired curve, and P_m^c is the corresponding point on the current curve Using (5) it is thus possible to obtain a measure of the diffence between the two curves that is independent of the way the points on the desired curve are generated, and on the distribution of the points on the coupler curve. The measure χ is chosen as the objective to minimize in order to redesign the mechanism so that it generates the desired curve

3 3 SENSITIVITY ANALYSIS

In the optimization procedure the derivatives of the objective function with respect to design variables are needed. The design variables may be any dimension of the mechanism that has an influence on the generated curve, such as link lengths, positions of joints, orientation of joint axes, etc. In this work the derivatives are approximated by numerical diffences, so that

$$\frac{\partial \chi}{\partial b_k} \approx \frac{\chi(\mathbf{b}+\Delta\mathbf{b}_k) - \chi(\mathbf{b})}{\Delta b_k} \qquad (6)$$

in which $\mathbf{b}$ is the vector of design variables, b_k is the k'th element of this vector and $\Delta\mathbf{b}_k$ is a vector in which all entries except the k'th are zero The k'th entry is equal to Δb_k, which is "a small quantity". In the examples for which the procedure have been employed, a value of .001 has appeared resonable.

By implementing the derivatives this way the calculation of the gradients is easily implemented, since only a pertubation of one of the dimensions and one analysis of the mechanism is needed for each gradient. It is thus again the general analysis program that is needed to do the calculations. The draw-back is that the computer time spent in the synthesis phase grows linearly with the number of design variables.

3 4 THE INTERESTING PERIOD

In section 3 2 it was assumed that the first and last point on the current curve were known This is seldom the case and it is desirable to avoid having to specify that

In this work this is avoided by specifying extra design variables in addition to the ones defined in section 3.3. The first is in relation to the initial position of the drivers Take e g. a revolute joint with a driver that prescribes a constant angular velocity. It is given by

$$\theta_j = \omega t + \phi_0 \qquad (7)$$

Here, ϕ_0 determines the position of the driven body at time equal zero, and hence the position of the coupler point at the initial time. Thus, introducing ϕ_0 as a design variable any restrictions on the initial point on the curve are removed Similar variables may be introduced for other drivers. Following the same line, the period in which coupler points that should be included in the comparison with the desired curve may be introduced as a design variable, defining what is here referred to as "the interesting period"

4. Continuation method

The above mentioned procedure works well if the initial guess creates a curve that is close to the desired one With poor initial guesses, though, the method may not converge One way to remedy this is to use a continuation method, as used in e.g. [8] or [11] It is employed as follows: Between the curve generated by the initial mechanism and the desired one, a number of curves are determined by interpolation between the two given ones. The one closest to the initial curve is selected to be the current desired curve, and the optimization procedure is applied in order to design a mechanism that generates this curve. Then the next curve is selected as the desired curve etc until the original desired curve is selected, and a mechanism determined that is capable of generating the curve

Again, the two curves need not be described by the same number of points, and hence spline interpolations are being performed on both curves, and the interpolations between the two curves are performed on the same number of points on each of the curves. It is thus possible to generate straight lines between corresponding points on the curves and perform a simple linear interpolation between these points

5. Examples

Below, two examples are given in which the procedure has been employed In the first example, a spatial slider-crank is synthesized to generate a straight line as part of its

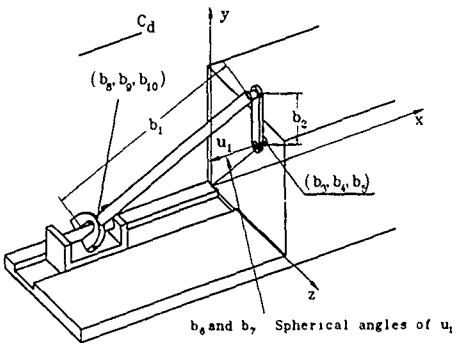


Fig 4 The design variables of the spatial slider crank It is shown with the values of the initial design

coupler curve, and the second is the spatial Geneva wheel, also found in [11], in which a spatial four bar mechanism is sought designed so that it can drive a spatial Geneva wheel.

5.1 THE SPATIAL SLIDER-CRANK

Consider the spatial slider crank shown in Fig. 4. The goal is to have part of the coupler curve generated by a point on the coupler to follow the straight line shown as C_d. The design variables are shown in Fig. 4, too. They are the lengths of the coupler, b_1 and the crank b_2, the location (b_3, b_4, b_5) of the revolute joint that the crank rotates about, the direction, given as the two spherical angles b_6 and b_7, of this joint axis, u_1, and the coordinates of the coupler point (b_8, b_9, b_{10}). In addition to that the two conditions for the interesting period are used as design variables, giving a total of 12 design variables. The desired straight line is given by 10 points along the line from (-0.05, 0.20, -0.10) to (-0.15, 0.20, -0.10). The initial design is given by

$$\{b\}^T = \{\ 0.30,\ 0.10,\ 0.00,\ 0.12,\ 0.10,\ 0.0,\ 0.0,\ 0.05,\ 0.05,\ 0.05,\ 0.05,\ 0.30\} \qquad (8)$$

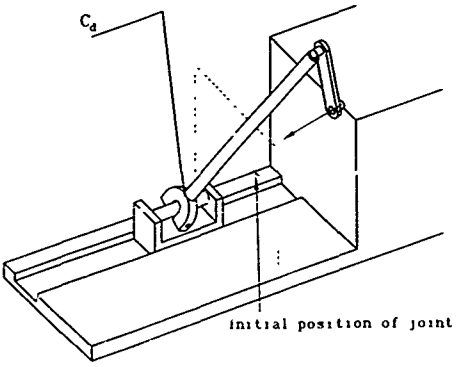


Fig 5 The final design of the spatial slider crank

The synthesis was carried out using 4 continuation steps. In each continuation step an average of 15 iterations was needed in the optimization process, and the final design

variables were

$$\{b_i\}^T = \{\,0.284,\ 0.073,\ 0.056,\ 0.070,\ 0.209,\ 1.776,\ 0.300,\ -0.197,\ -0.046,\ 0.056,\ 0.171,\ 0.115\} \tag{9}$$

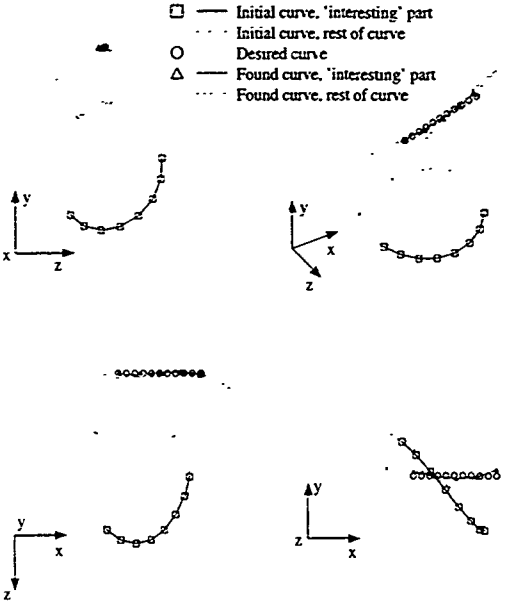


Fig 6 The initial, the desired and the resulting curve for the spatial slider-crank mechanism, shown in four view directions

At the end of the iterations the objective function had a value corresponding to an average deviation of about 1 mm between the two curves. The resulting mechanism is shown in Fig 5 In Fig. 6, the initial coupler curve, the desired one, and the one generated by the procedure are shown.

5.2. THE SPATIAL GENEVA WHEEL

This example may also be found in [11], where the synthesis is performed using a procedure specifically designed for the spatial four bar mechanism with four revolute joints.

The goal is to generate a spatial four bar mechanism with four revolute joints, that can drive the spherical geneva wheel shown in Fig. 7.

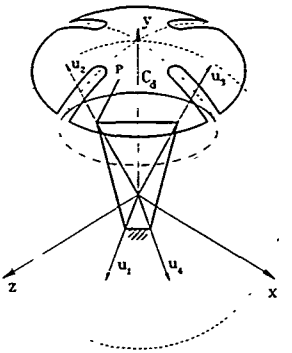


Fig 7 The spherical geneva wheel.

From [11] it is known that if the four revolute joints are placed on the sphere, then it is possible to design a spatial four bar mechanism that can drive the wheel The path that the coupler point must follow in order to drive the wheel is given in [11] as

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} \cos\gamma \sin\alpha \\ \sin\gamma \\ \cos\gamma \cos\alpha \end{Bmatrix} \tag{10}$$

with

$$\alpha = -\frac{1}{4}\sin(2\pi\tau) - \frac{\pi}{2}\tau$$
$$\gamma = -\frac{3}{2}\tau^2 - \frac{3}{2}\tau + 0.927 \quad , \quad 0 \le \tau \le 1 \qquad (11)$$

Based on (10) and (11) a number of points can be generated to define the desired curve.

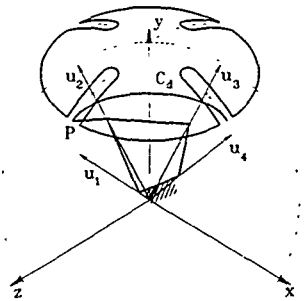


Fig 8 The final design of the Geneva wheel

Since the joints are restricted to lie on the unit sphere with the joints axes being directed from the centre of the sphere pointing radially out through the joint points, both the position of the joints and the direction of the joint axes are defined through the two spherical angles of the joint axes. Hence, the design parameters are the two spherical angles for each joint and the coordinates of the coupler point, shown as P in Fig 7, and the two interesting period parameteres, giving a total of 13 design variables The initial configuration is the same as the one given in [11], which corresponds to the design variables being

$$\{b\}^T = \{0\ 8633.0\ 4326.1.204.0\ 9800.0.3661.0.9795.0.7074$$
$$0\ 4326.0\ 0000.0\ 0000.0.0000.0\ 0000.0\ 3000\} \quad (12)$$

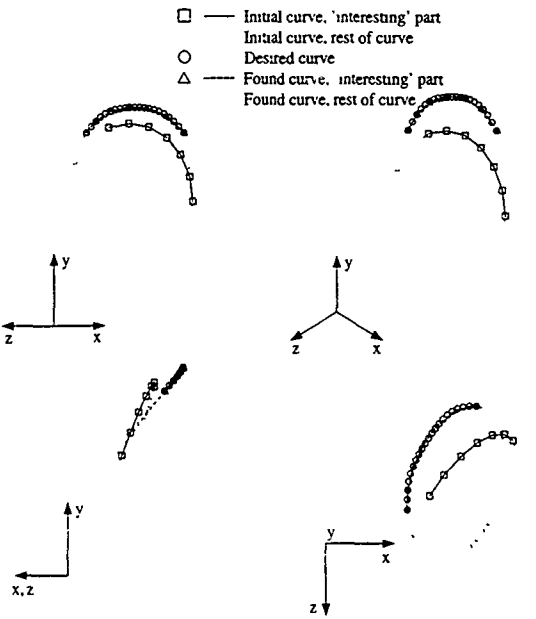


Fig 9 The initial, the desired and the final curve for the Geneva wheel, shown in four view directions

The optimization procedure was again applied using 4 continuation steps Here, though, the linear interpolation between the curves was not used. Since the curve is known to lie on the sphere, the interpolations were made on arcs between the initial and desired curves to ensure that the intermediate curves were on the sphere, too

In this example the optimizer needed an average of 23 iterations in each step, and the final values of the design variables were

$$\{b\}^T = \{0\ 8633,0\ 6516,1.254,1\ 013,0\ 3722,0\ 9902,0\ 5643,\ 0\ 7269,0.1255,-0\ 1164,-0\ 2166\ 0\ 8607,0\ 2971\} \quad (13)$$

The resulting mechanism is shown in Fig 8, and the initial, the desired and the generated curves are shown in Fig 9 In this example, too, the value of the objective function corresponds to a deviation of about 1 mm from the desired curve.

6. Conclusion

In this work a general purpose tool has been developed with which it is possible to synthesize the path generation for arbitrary spatial mechanisms. The method is based on a general purpose analysis method It is therefore not restricted to particular mechanisms, but is capable of synthesizing any mechanism that can be analyzed with the analysis program Also, the desired path is described by a number of discrete points, and it is therefore possible to define arbitrary paths as desired curves

Introducing the "interesting period", i.e. the position of the input links at initial time and the interesting time, as design variables, the need to specify which part of the coupler curve that is the pathgenerating part has been eliminated.

To improve the stability properties in the procedure a continuation method is implemented I e , the procedure does not attempt to generate the path directly from the initial guess on the dimensions, but instead the desired curve is gradually changed towards the actual desired one in a number of steps, and an optimization procedure is applied in each step to minimize the difference between the two curves

At present, the sensitivity analysis is performed with finite difference approximations of the sensitivities. The major draw-back of that is that the method is very slow, since a sensitivity calculation requires a complete analysis for each design variable A natural continuation of this work is to implement analytical sensitivities in the procedure

7. References.

[1] Erdman, A. G and Sandor, G N.: "Mechanism Design Analysis and Synthesis", Prentice Hall, New Jersey, 1991.

[2] Erdman, A G and Sandor, G N "Advanced Mechanism Design Analysis and Synthesis", Prentice Hall, New Jersey, 1991.

[3] Haug, E J. and Sohoni, V N . "Design Sensitivity Analysis and Optimization of Kinematically Driven Systems", in Computer Aided Analysis and Optimization of Mechanical System Dynamics, NATO ASI Series, Ed · E J Haug pp 499-554

[4] Kramer, S N and Sandor, G N . "Selective Precision Synthesis - A General Method

of Optimization for Planar Mechanisms", Journal of Engineering for Industry, 1975

[5] Root, R R., Ragsdell, K. M. "A Survey of Optimization Methods Applied to the Design of Mechanisms", Journal of Engineering for Industry, 1976

[6] Starns, G. and Flugrad, D. R.: "Five-Bar Path Generation Synthesis by Continuation Methods", Advances in Design Automation, 1991.

[7] Subbian, I., Flugrad, D. R. "Four-Bar Path Generation Synthesis by a Continuation Method", Advances in Design Automation, 1989

[8] Tsai, L. W and Lu, J. J. "Coupler-Point-Curve Synthesis Using Homotopy Methods", Advances in Design Automation, 1989

[9] Tsai, L W and Morgan, A P. "Solving the Kinematics of the Most General Six- and Five-Degree-of-freedom Manipulators by Continuation Methods", Journal of Engineering for Industry, 1984

[10] Hansen, M. R. "An Automated Procedure for Dimensional Synthesis of Planar Mechanisms", Structural Optimization, Feb. 1993

[11] Angeles, J and Liu, Z: "A Constrained Least-square Method for the Optimization of Spherical Four-bar Path Generators", in Mechanism Synthesis and Analysis, 1990, Ed 's: McCarthy, M., Derby, S. and Pisano, A.

[12] Chiang, C S., Cheng, W. H. and Hoeltzel, D. A "Synthesis of the RSCR Mechanism for Four Precision Points with Relaxed Specifications", in Mechanism Synthesis and Analysis, 1990, Ed.'s McCarthy, M., Derby, S and Pisano, A

[13] Nikravesh, P E. and Gim, G.. "Systematic Construction of the Equations of Motion for Multibody Systems Containing Closed Kinematic Loops", Paper no 89-DAC-58, Proc. ASME Design Automation Conference, Montreal, Canada, Sept. 17-20, 1989

[14] Nikravesh, P E "Computational Methods in Multibody Systems", Lecture notes from COMETT course on Computer Aided Analysis of Mechanical Systems, Lyngby, May 1991

[15] Nikravesh, P E "Computer Aided Analysis of Mechanical Systems", Prentice Hall, 1988

DYNAMIC SIMULATION IN WEAPON SYSTEM DESIGN

Robert M. Dombroski
U.S Army ARDEC
SMCAR-FSA-F Bldg 61N
Picatinny Arsenal, NJ 07806-5000

ABSTRACT. Artillery Armaments Division of the Fire Support Armament Center (FSAC), has been involved in the mechanical computer aided engineering (MCAE) of weapons systems since 1987 During this time Artillery Armaments Division has become FSAC's acknowledged experts in computerized weapon system and munitions design and analysis Artillery Armaments Division's capability currently encompasses all aspects of computer aided design, conceptual solid modeling, finite element modelling and analysis, dynamic weapon system analysis, computer aided manufacturing analysis and computer aided drafting. Artillery Armaments Division is at the forefront of computer hardware, software and networking technology.

This paper will discuss the development of a liquid propellant (LP) armament system for a self propelled howitzer. The LP weapon system is being developed as one of the future weapon concepts under the Advanced Fire Support Armament System program. The LP effort demonstrates the latest in all MCAE fields, conceptual design, finite element modelling and analysis, computer aided drafting, and computer aided dynamic simulation and analysis This paper will focus on the dynamic modelling, simulation and analysis used in the design and production of a prototype weapon system as well as the key role that analysis plays in the design cycle.

Weapon System Design

In the past, it took the U.S Army approximately 15 years for request to delivery on much of its fielded equipment More recent figures indicate a 10 year lead time This is unacceptable The Fire Support Armament Center (FSAC) of the U.S. Army s Armament Research, Development and Engineering Center (ARDEC), Picatinny Arsenal, NJ, has increased its use of MCAE technology to dramatically reduce design, development, and time to production by one third

Typically, the design cycle is a chaotic and undocumented process at worst, and a slow, sequential process at best, with progress measured by the "thrown-over-the-wall-to- ." milestone. This leads to a 'build-test-fit' development cycle where the final product design is dictated by whether or not it is time for the design phase to end FSAC took a three phased approach to the problem of a sequential, poorly defined design effort. First, FSAC implemented a distributed computing architecture which support synchronization, optimization and refinement of the information flow. Next FSAC adopted a unified representation of the design itself, which could be accessed by all the elements of a concurrent engineering team. Finally, FSAC implemented an integrated set of MCAE software which streamlined the design/analysis process This effort resulted in an electronic capability which not only enables concurrent engineering, but encourages design by analysis of future weapon systems

Liquid Propellant Design Program

The Artillery Armaments Division (AAD) is FSAC's primary artillery weapon system research and development organization. AAD has taken the lead at ARDEC in both MCAE capability and concurrent engineering AAD's mission is to provide total lifecycle systems engineering of artillery systems. AAD is responsible for planning, directing, performing and controlling the research, conceptual design, engineering development, product engineering, product improvement and producibility engineering for material including light and medium towed and air mobile artillery, medium and heavy self-propelled artillery, mortars, recoilless rifles and their munitions AAD is also responsible for the conceptual design, analysis and preliminary development of future weapon systems which will improve the firepower of the U S. Army

One such project is development of a liquid propellant cannon for use in self propelled artillery systems This cannon has many advantages over solid propellant weapons such as reduced manufacturing costs, less waste in fielding charges, and less labor to handle the ammunition. The armament system is being designed by General Electric in parallel with the Artillery Armaments Division and Benet Labs of ARDEC who are designing and fabricating the gun mount and the breech mechanism respectively as well as modifying the vehicle to accept the new system Using Pro Engineer software from Parametric Technologies Corporation and Patran Plus from PDA Engineering and IDEAS from SDRC Inc designers from AAD, located at Picatinny, and Benet Labs, located at Watervliet NY, were able to create their respective parts simultaneously. Dynamic analysis using DADS from CADSI Inc. was performed by analysts at Picatinny on the completed assembly.

Its is difficult to put a value on the time that was saved using an integrated MCAE design and analysis capability. However, the design was completed at least 30% faster than previous projects, and that includes many more design iterations and analysis permutations than would have been possible in the past. Other benefits came downstream in the process from attaining a more fine tuned design fit, that more closely meets specifications As a result, time typically required for the knock-down, reanalyze, redesign, rebuild and retest

cycle was cut significantly

WORKING IN PARALLEL

In FSAC's concurrent engineering efforts, the software selected not only automates the engineering tasks, it actually allows FSAC to change their workflow to better facilitate communications, teamwork and early involvement. In FSAC, the design/development process has been re-engineered to take full advantage of our MCAE tools FSAC's mechanical design efforts center on the solid model. The main reasons for this are the solid model's ability to enable all team members to visualize the product from the earliest stages of the design phase and the fact the the solid model contains all the physical characteristics of the product, making it much more accurate than wireframe or drawings

Once detailed parts of the liquid propellant system were completed, subassemblies and the main assembly were analyzed. Here, many minor revisions were made to each part, adjusting chambers, bolt holes, and mating surfaces for fit. First cut drawings were generated as designers finished each part, and electronically forwarded to draftsmen who created final drawings. A solid model view of the Liquid propellant Gun System is shown in figure 1

ANALYSIS

In FSAC's concurrent engineering environment, analysis drives the design as opposed to validating it. As a result, the design team is provided with early and continuous feedback on many or all aspects of the design, including such things as performance, survivability, reliability and manufacturability. Inconsistencies, design flaws and so on are trapped earlier in the design phase which greatly reduces expensive and time-consuming redesign later in the development process.

Analysis on various elements of the gun mount, cradle and cab began as soon as models had been roughed out. Deflection in the cradle was examined using Swanson Analysis Systems Inc.'s ANSYS finite-element analysis software, as was a static analysis of the loads on the mounting point's lifting cylinders. Anticipated loads on the cab were determined with CADSI's DADS software. Results were post-processed in Patran where several areas of unacceptable stress and deflection were indicated, prompting modification to original designs early in the design process. Typically, the need for these changes would not have been discovered until a prototype was built and tested.

SIMULATION IN THE DESIGN PHASE

Historically, when modelling and redesigning military systems and subsystems such as an automatic loader design engineers had two alternatives. The first and most often chosen was not to design a new item but to adapt existing subsystems to meet new requirements The second alternative was for an engineer to write specialized FORTRAN codes to

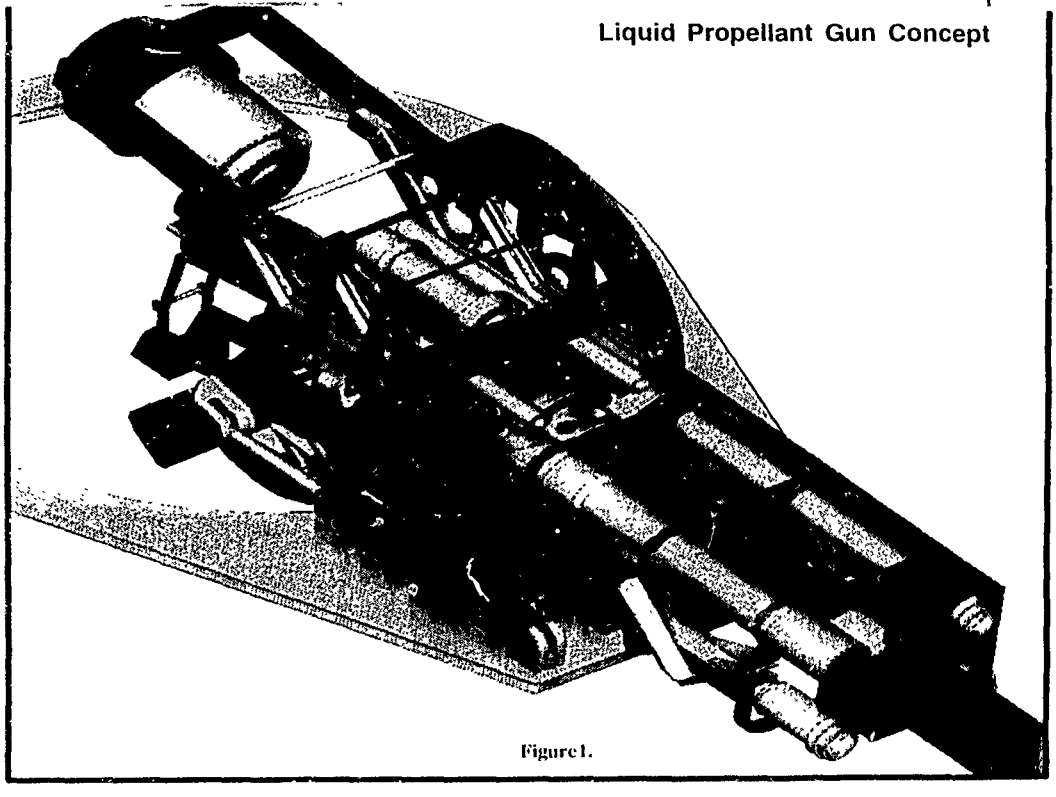


Figure 1.

perform specific analysis functions

Advanced hardware and software technology is now providing engineers with a third option, to design new systems without writing custom equations of motion but rather to utilize motion analysis software This technology allows designers to enhance existing engineering designs through analysis results and realistic visualization

DADS, the Dynamic Analysis and Design System, was used to model the LP3 gun mount and automatic loader mechanism as well as to design and analyze the crucial breech control cam system DADS allows designers to quickly and effe..ively address complex questions and analyze design performance. Dynamic and kinematic 'ysis is used to understand how the individual parts within an assembly move and what forces they endure when the assembly interacts with external forces.

The Breech Mechanism

The breech is made up of two parts; a breech camplate and a crank The crank moves along two campaths The first campath controls the locking and unlocking of the breech while the second campath actually swings the breech into the open position. The motion of the gun tube during recoil activates rollers which move on the campaths and drive the motion of the crank. As the roller moves along the first campath it unlocks the breech mechanism After a brief delay the second roller is engaged As the roller moves along, the breech door swings open to allow loading of the projectile and munition This motion is depicted in Figure 2.

The objective of the motion analysis was to alter the two campath so that the breech door would open without interference An existing loader system was used as a baseline design. This loader was inadequate because of an interference between the autoloader, breech and gun tube.

The modelling of the breech mechanism focused on 2 dimensional kinematic analysis For analysis, the mechanism was broken into individual bodies This breech model consists of two bodies, the breech camplate and the crank. Data describing these bodies and their interconnection was entered with a menu-driven preprocessor An on-line input data error checking facility immediately detects most common errors such as improper boundary conditions or connectivity

After creating the bodies, the designer defined the joints using a library of joint elements. To complete the cam-follower joint definitions, the designer also specified the initial contact angle. This is an estimate of the initial angle of contact and the point on the body which is to follow the track at all times. The engineer then defines a driver, which in this case was a simple polynomial The driver defines the x motion of the crank's pivot point

Once these elements were created, the data was submitted for analysis. The analysis program automatically assembles the model to satisfy constraints imposed by the defined joints The program then constructs the equations of motion and estimates of the initial location and orientation of the bodies. The solutions to the equations are found by using a variable order time-step predictor-corrector algorithm

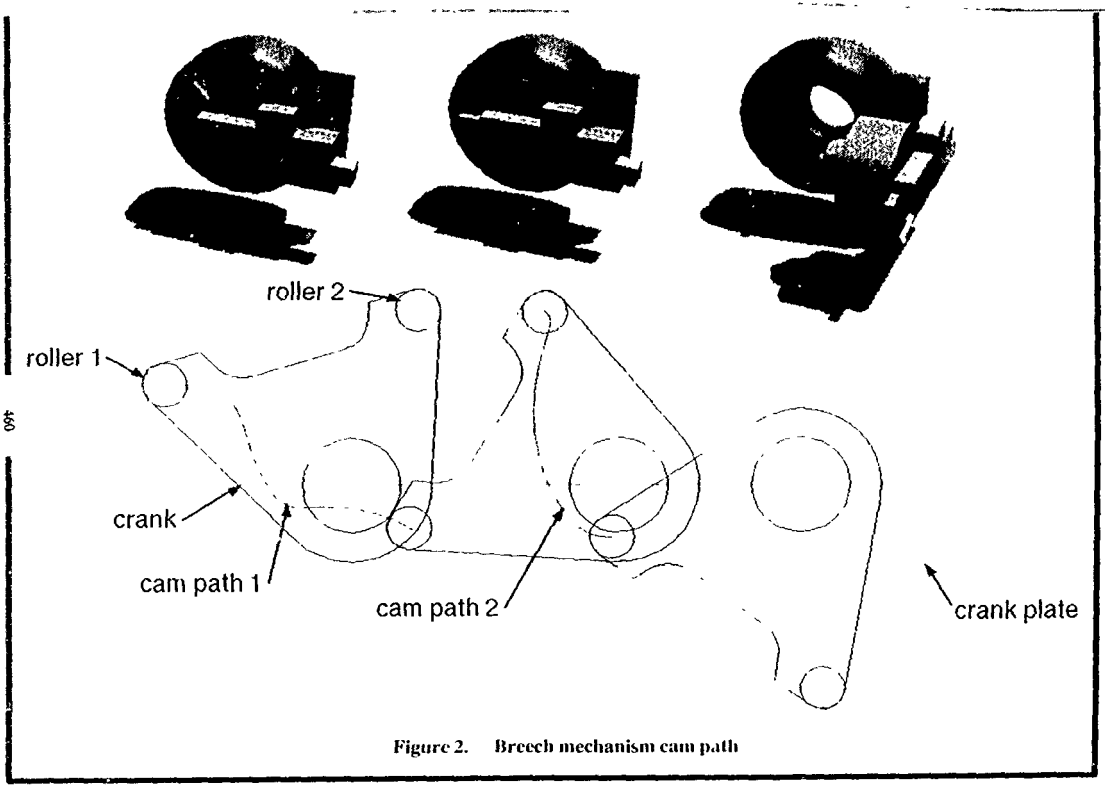


Figure 2. Breech mechanism cam path

The designer then examined the results. The interpretation of results is the most important phase of the analysis process. It provides immediate verification of whether the initial design meets the specifications as entered. The analysis results include the positions, velocities and accelerations at the centers of gravity of all bodies and other chosen points

VISUALIZATION

The full visual impact of the breech door and the entire loader mechanism in motion was made possible through sophisticated simulation capabilities of the analysis program. Designers were able to realistically see the crank successfully move through the campaths and the motion of the breech door and autoloader mechanism during the firing cycle. After the initial modelling of the breech mechanism and the loader was developed and analysis performed, a detailed animation was created by adding more complex geometry. The full animation depicts the gun tube and other components in detail, and depicts the entire armament system as it functions through an entire load, fire and recoil cycle. Depending on the designers' purpose, the simulation and animation of the design can be simple or complex. A simple model is used for initial visual confirmation, while a complex model can be developed for presentation purposes. A sample of the DADS simulation model is shown in figure 3

BENEFITS OF SIMULATION

The simulation of the autoloader and breech cam was invaluable to the design process. In addition to enabling the campath design and analysis, it visually depicted all the kinematic relationships between the recoiling parts, the cradle, breech, ballistic shield, autoloader and the vehicle turret. Without this, space claim and interference checking during system dynamics would cause design milestones to be adversely impacted. The animated model communicated the mechanical concepts involved in the design to all involved in the project.

Also, from the analysis point of view, the dynamic data provided force-load information which was critical to to the determination of the strength of the cradle structure. The interactivity of DADS and the visualization capabilities enabled the designer to examine joint and body locations and body masses. These he could quickly modify to generate new force load curves for analysis.

Concurrent Engineering

The Liquid Propellant Gun system design effort was a collaboration of designers, engineers and analysts from both the federal and private sector. The Liquid propellant armament design was done by General Electric in Pittsfield, MA. The breech mechanism was designed at Watervliet Arsenal, NY. The gun mount was designed at Picatinny Arsenal in Dover, NJ. The overall system integration was performed at Picatinny Arsenal.

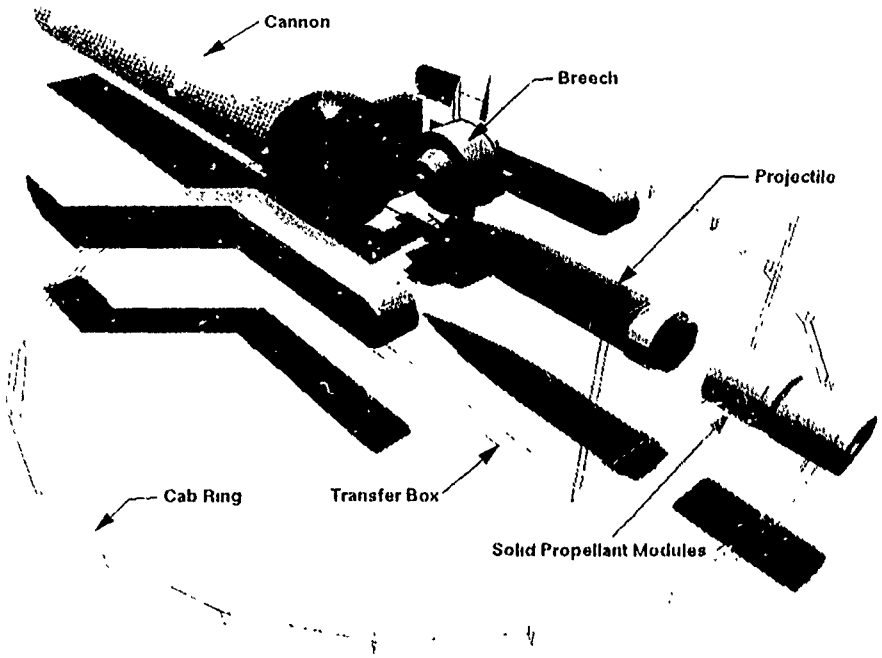


Figure 3. Simulation of howitzer with autoloader

Design and analysis efforts were performed simultaneously at all three locations. All design data and parameters was accessible at each location via electronic data transfer of solid model geometry, thus allowing each subsystem designer to know how their portion of the effort interacted with the others. Design changes which impacted other subsystems were quickly identified avoiding costly and time consuming problems later on in the development cycle. Knowing the operational characteristics, dimensions and space requirements of all the subsystems greatly added to the accuracy of the analyses performed on the system.

Internally at ARDEC, the Liquid propellant weapon system development effort was comprised of a wide range of engineering disciplines. The ability for designers, analysts, simulation specialists, draftsmen and program managers to function both independently, yet in an integrated and concurrent manner could only be accomplished through electronic means. AAD's design and analysis data as well as all its MCAE software resides on various mainframe file servers. Utilizing the latest networking techniques, AAD is able to make this data appear to each user as a single shared corporate data base, residing on his workstation. Project management software embedded in Pro Engineer orchestrates overall security and access to common data.

BIOGRAPHY

Robert M. Dombroski is a team leader in Artillery Armament Division, Concept and Analysis Branch at ARDEC. In addition to providing technical supervision for a multi-disciplinary analysis team he is responsible for all the physical operational aspects of AAD's MCAE capability. Mr. Dombroski has been responsible for the total system design, specification and implementation of AAD's MCAE system.

Previously, Mr. Dombroski was the lead engineer for robotic systems integration on an autonomously functioning self propelled howitzer.

Mr. Dombroski is a graduate of St John's University, Jamaica, NY with a Bachelors degree in Mathematics and has a Masters degree in Systems Management from Florida Institute of Technology, Melbourne, Florida. He has recently received the ARDEC Commander's Award for Civilian Service and the St. Barbara's Medal for contributions to advancements in the area of field artillery.

Modelling of Automotive Vehicles for Motion Control Studies

A. COSTA	*R.P. JONES*
EESC - USP	*University of Warwick*
Av. Carlos Botelho 1465	*Gibbett Hill Rd.*
13560 Sao Carlos	*CV4 7AL Coventry*
SP Brasil	*England*

Abstract

The subject of this paper is the application of multibody system techniques for vehicle chassis modelling aimed at the development of integrated vehicle control. It realises that resulting models can be complex and that simplifications in chassis description is recommendable. For this purpose, it has developed a technique for representing suspension geometry effects, which by taking the MBS structure into account, results in small and fast runtime simulation models. Yet, the model is capable of describing the full range of normal operation of the automotive vehicle. Using the previously developed model, comprehensive analysis of all aspects of vehicle motion is carried out. The objectives of such analysis is the determination of a driving envelope in which the use of linearised models of the nonlinear chassis can be justified for control analysis and design.

1 Introduction

The period following the early 1970's has seen a rapid increase in the application of automatic control techniques in automotive vehicles. These past and current developments in automotive vehicle control have progressed mostly in a piecemeal fashion, whereby individual vehicle subsystems, such as the engine, suspension and braking systems have been studied in isolation. Future applications of control in automotive vehicles will follow a trend towards system integration, leading ultimately to the development of integrated vehicle control systems capable of coordinating the action of the various subsystems. The coordination and integration of automotive vehicle subsystem control requires the interaction amongst the various subsystems to be taken into consideration at the control design stages, i.e., a total system approach to automotive vehicle control is needed.

Those aspects of vehicle motion control which are of primary interest span a dynamic range with a frequency cut-off below 50 Hz. Up to this frequency, simulation studies of automotive vehicle dynamics can justifiably be based on lumped parameter models [9]. Current approaches to vehicle dynamics simulation fall into 2 distinct categories. The first approach involves the use of simple mathematical models which are derived from first principles and which are usually assembled manually.

The second approach to vehicle dynamics simulation involves mathematical models which incorporate representations of the vehicle kinematics, tyre characteristics and suspension geometry effects. This approach to vehicle dynamics simulation can result in mathematical models which are highly complex. Consequently, automated model generation facilities based on multibody systems (MBS) modelling techniques are widely employed.

Approaches to MBS modelling based on a combination of Kane's Method [4] and symbolic computation has been shown [3] to generate the most efficient (run time) simulation code.

While the use of those more powerful and complex models makes the idea of an integrated control system very difficult due to the high nonlinearities of the equations of motion as well as the great number of degrees of freedom; it has not been answered yet whether a simplified model, however taking some relevant interactions from a control point of view, into consideration, could result in improved vehicle motion. A positive answer to such a question would possibly mean, for example, a reduced number of transducers and their related apparatus necessary to be implemented in a future vehicle in which these concepts might be applied.

2 Modelling of an Automotive Vehicle for Motion Control Studies

Apart from aerodynamic resistances and gravitational effects, all external forces acting on a vehicle are applied through the wheels. Consequently, total vehicle control will entail the application of control action at the wheels, and will be based on a combination of propulsion, steering and suspension control. Therefore it is necessary to develop vehicle models which are capable of representing all these actions simultaneously.

The modelling of this work is based on an approach [2], in which suspension geometry effects are incorporated in a way which does not involve a detailed representation of the suspension system. The basis of the approach is an empirical black-box representation of the suspension kinematics derived from readily available experimental data describing the suspension trajectory at each wheel hub and wheel orientation. From this data, an equivalent swing axle or trailing arm model is derived using techniques from differential geometry.

The simulation model is constructed using the MBS package SD/FAST [10], and the ACSL [1] continuous simulation language. SD/FAST combines Kane's formulation for a MBS model with an equation generator incorporating a transparent symbolic computation facility, while ACSL provides the simulation environment necessary to solve these equations.

2.1 Schematic Description of Vehicle Model

A schematic view of the nonlinear chassis model is presented in figure 1. The model represents the kinematics of the sprung mass and the four wheels and incorporates the effects of geometrical constraints associated with the suspension

At each wheel, the suspension geometry effects are represented in a black box manner as a swing axle $i3_i$ connecting the wheel hub H to the sprung mass via a single degree of freedom rotational joint (pin joint) at the point. The pin position and the arm length of the swing axle are derived empirically from data defining the wheelbase/bump and track/bump characteristics at each wheel hub. The prescribed motion is incorporated in the simulation code and imposed to the appropriate joints.

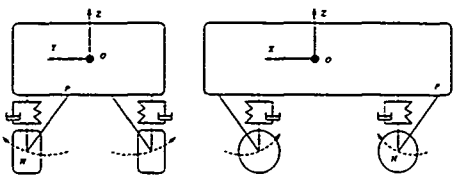


Figure 1 Nonlinear Chassis Model

In all cases, the polynomial of lowest order which adequately fitted the data was chosen. A comparison of these polynomial representations and the corresponding wheelbase and track characteristics data is presented in figures 2 and 3.

The variation in fore-aft and lateral wheel displacements described in these figures appears small. However, the effect of these variations on the vehicle pitch and roll characteristics is highly significant, particularly during vehicle manoeuvres associated with large acceleration levels.

The model provides a ten degree of freedom representation of the chassis, with six degrees of freedom resulting from the sprung mass and one degree of freedom from each swing axle. This results in a nonlinear twenty state model, in which the states correspond to ten generalised coordinates and ten generalised speeds. This swing axle model provides the simplest possible representation of the suspension system which is consistent with a nonlinear chassis model intended for motion control studies. This chassis model represents the first stage in the development of a vehicle dynamics model for use in motion control studies. The chassis model developed in this chapter is based on data corresponding to a luxury European saloon car. This data was selected for illustration purpose only. In particular, it should be noted that the chassis modelling approach presented here is not restricted to luxury saloon cars but is general and applicable to a wide variety of automotive vehicle types.

2.2 Topological Representation of MBS Model

The chassis model is considered as a multibody system made up from five lumped connected rigid bodies organised in a tree topology as described in figure 4. The central, or base body corresponds to the sprung mass and the four branch bodies to each of the four wheels and associated suspension links.

Associated with the base body and each branch body is a local right-handed

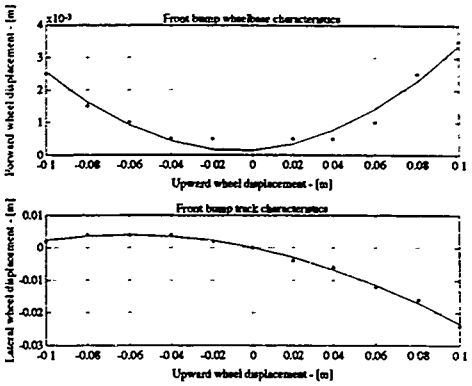


Figure 2: Front Wheelbase and Track Characteristics

coordinate frame with fixed orientation and location (at the centre of mass) in the respective body.

The geometrical description of the MBS model can be specified by a set of three vectors for each of the branch bodies The first of the three vectors *op* describes the position of the pint joint relative to the centre of gravity of the base body This vector is fixed in the base body The second vector *hp* describes the position of the pin joint relative to the centre of gravity of the branch body and is fixed in the branch body The third vector *b* describes the orientation of the pin joint axis

		Left	Right
F	op	(1.441, 0.303, -0.242)	(1.441, -0.303, -0 242)
	hp	(0.121, -0 447, 0 058)	(0.121, 0.447, 0.058)
	b	(0 966, 0.257, -0 029)	(0.966, -0.257, -0.029)
R	op	(-1.516, 0 105, -0 157)	(-1.516, -0.105 -0.157)
	hp	(-0 016, -0 645, 0.143)	(-0 016, -0 645, 0 143)
	b	(0.994, 0 0, 0 113)	(0.994, 0 0, 0 113)

Table 1 Geometrical description of MBS chassis model

For each branch body, specifications for the vectors *op*, *hp* and *eb* were obtained relative to this reference coordinate frame The vectors *hp* and *b* were derived using differential geometry and the vector *op* was derived from the vehicle dimensional data presented in table 2. The resultant values of the vectors *op* *hp* and *b* are presented in table 1 To complete the physical description of this

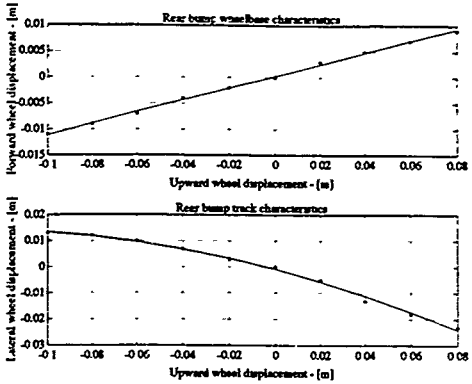


Figure 3 Rear Wheelbase and Track Characteristics

MBS model, the mass and inertia matrix for each body is required. This data is also presented in table 2

Effective weight of wheel	front	50 kg
	rear	40 kg
Radius of wheel		0.2 m
Weight of sprung mass		1600.0 kg
Principle inertias of sprung mass	I_{xx}	500.0 $kg.m^2$
	I_{yy}	3000.0 $kg.m^2$
	I_{zz}	3000.0 $kg.m^2$
Centre of gravity of sprung mass	from front axle	1.32 m
	from rear axle	1.5 m
	above ground	0.5 m
	from right/left side	0.75 m

Table 2. Vehicle inertia and geometric data

2.3 External Forces Acting on the MBS Model

Apart from gravitational influences, the forces acting on the sprung mass are aerodynamic and suspension forces, which also includes the effect of an antiroll torsion bar at the front axle. Suspension characteristics are given in table 3 Aerodynamic forces and torques have also been modelled, but are not described in this paper

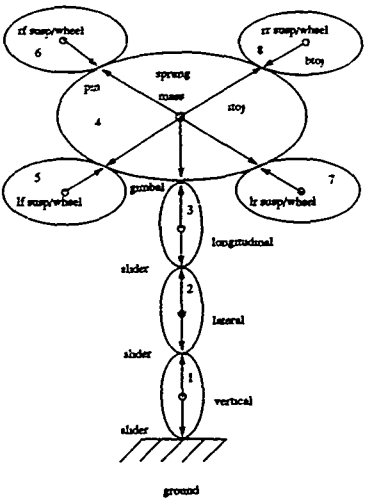


Figure 4: Topological Representation of Chassis

The forces acting on each unsprung mass are the longitudinal, lateral and vertical tyre forces, the suspension forces and gravitational forces. Tyre characteristics are also described in table 3. The suspension and tyre forces are described by the parameters given in table 3.

Each suspension force is modelled as a spring in parallel with a damper. The spring and damper forces are taken as linear functions of the spring displacement, and displacement rate, respectively, relative to the base body. An additional roll torque represents the effect of an antiroll torsion bar acting at the front axle. This torque is modelled, relative to the base body, as a linear function of the left/right difference in spring displacements at the front axle. Static values for the vertical suspension and tyre deflections are computed to balance the weight of the sprung mass in the steady-state.

For the linear tyre model, the longitudinal tyre forces represent propulsion and braking action. These forces are modelled as an externally defined forcing term, where it is assumed that wheel spin and wheel lock do not occur. The vertical and lateral tyre forces are represented as linear functions of the vertical wheel displacement and wheel's sideslip angle, respectively.

2.4 Simulation Model of Chassis

The computer simulation model was built using the MBS modelling package SD/FAST and the ACSL continuous simulation language The SD/FAST program processes an ASCII source code file containing a user-supplied description of a MBS model. An example of part of the program is given below.

The resulting source code, which is generated autonomously by SD/FAST, consists of two main FORTRAN subroutines and several secondary subroutines Approximately 10 s of CPU time on a SUN 4/330 computer was used to generate the resulting output file. This output file consists of around 1500 lines of FORTRAN code, about 70% of which is contained in the main subroutine defining the state derivatives for the equations of motion of the MBS model

The approach to simulation model development for an automotive vehicle chassis system, described in this section, has the advantage that the model can be easily extended to include nonlinear tyre and suspension force characteristics without the need to modify the kinematic description of the sprung and unsprung masses. More importantly, the approach facilitates the development of vehicle models which combine a MBS description of the chassis with other vehicle subsystems which are not amenable to MBS modelling techniques, such as the powertrain or a digital control system, but which can be easily represented in a simulation language such as ACSL

2.5 Model Validation

T he amount of experimental data available was very limited, and therefore extensive validation was not possible to be performed, as it would be desirable The only experimental results available consisted of the static deflections of the suspension, as it was used to derive the swing axle model Other results available were an eigenmode analysis resulting from simpler models in use by the vehicle s manufacturer and which had been previously validated by them.

The linear eigenmode analysis was performed with the computer simulation model as a prelude to nonlinear simulation experiments The natural modes of the MBS model and their corresponding frequencies and damping ratios were computed using the linear analysis facilities of ACSL. The linear analysis was carried out with the MBS model in a steady state condition corresponding to a

		Front	Rear
Susp.	Spring stiffness $kN.m^{-1}$	20 0	27 0
	Damper coefficient $kN\ s\ m^{-1}$	1 4	2.0
	Antiroll bar stiffness $kN\ m^{-1}$	20 0	
Linear tyre	Vertical stiffness $kN\ m^{-1}$	250 0	250.0
	Cornering stiffness $kN\ rd^{-1}$	66 0	70 0
Steering system compliance $rd.kN^{-1}$		0.0051	

Table 3: Suspension and linear tyre data

```
body        =  Right_front_wheel
inb         =  Sprung_mass
joint       =  pin
mass        =  50
inertia     =  0                    0                    0
inbtojoint  =  1.44090214828721     -0.30292586469092    -0.24179219374438
bodytojoint =  0.12090214828721     0 44707413530908     0.05820780625562
pin         =  0.96586483194595     -0.25738705686472    -0.02927506393730
```

constant forward speed of 20 m/s The eigenmode analysis resulted in four modes corresponding to the sprung mass and four further modes, two for each of the front and rear wheel pairs. The results were in close agreement with experimental observation and are presented in table 4

Modes	Frequency		Damping
	Hz	rd/s	Factor
Front-end bounce	1 1	6.88	0 206
Rear-end bounce	1 35	8 46	0 266
Roll on springs	1 81	11.38	0.267
Yaw rate/lateral velocity	1.29	8.13	0 834
In-phase front wheel hop	12 02	75.46	0.200
Out-of-phase front wheel hop	12.50	78.52	0 195
In-phase rear wheel hop	13.00	81.69	0 304
Out-of-phase rear wheel hop	13.05	81 95	0 303

Table 4: Eigenmode analysis

In the next section extensive simulations are carried out in order to determine an operating range and conditions for the use of a linearised model in control system analysis and design These simulations cover all aspects of vehicle motion under realistic operating conditions considering driver's inputs as well as external disturbances For that purpose transient and steady-state analysis is performed However only a few results are present for the reason of space Full details are given in [2]

3 Simulation and Analysis of Vehicle Motion

In order to confirm the utility of the previously developed model, as well as characterising its behaviour within an operating range which allows the definition of appropriate control strategies, for example, sets of of operating conditions in which linear approximations would still be valid, a large number of tests which exercise the model within the boundaries of its simplifying assumptions was carried out These tests or experiments, are defined to be either driver inputs or external disturbances and they will be looking at both transient or steady-state behaviour of the simulation model The analyses to be performed will consist

of a detailed study of these simulations and will include eigenvalue analysis intended at verifying the validity of transient behaviour and gain analysis, aimed at determining the characteristics of the model's steady-state characteristics

The intention of these experiments is to allow the definition of a driving envelope in which the interactions present, their nature and magnitude are discussed and a valid range of operating conditions for control studies can be defined

3.1 Vehicle Simulation

The simulation runs to be executed comprise two types of manoeuvre: one intended to analyse the vehicle's transient behaviour and the another aimed at obtaining the vehicle steady-state response characteristics However only a few results are shown in this paper for reasons of space The interested reader may refer to [2] for full details

3.1.1 Transient Manoeuvres

The simulation which is used to analyse the the transient behaviour of the model can be divided in two types: driver related inputs and external disturbances inputs. Driver's inputs will be in the propulsion system, either in acceleration or deceleration manoeuvres, and in the steering system, in cornering manoeuvre External disturbances will consist of road vertical input.

The steering input simulations which exercise the vehicle's handling or stability and steering control properties consist of manoeuvres of the so called 'free-control' type, in which steering wheel angle is considered as the input

Quantities depicted and analysed include yaw rate, lateral velocity and acceleration, roll angle, wheel slip angle, lateral and vertical tyre forces As it would be expected, the variation of other quantities of motion associated with vehicle s forward and vertical motion are negligible

For the acceleration and deceleration simulations, the powertrain and/or braking system dynamics are considered in a very approximate manner. Deterministic functions of time are adopted which account for typical time delays involved in these system's responses and the available level of performance for the type of vehicle being modelled. The input quantities will be either driving or braking forces for the linear tyre which assume no slip in the longitudinal motion of the wheels

The road vertical inputs consist of crossing obstacles of different types such that all relevant aspects of vehicle motion are exercised For this purpose, out-of-phase triangular bumps are transversed, together with terminated ramp and sinusoidal road profile inputs

The quantities of motion which undergo significant variation and are illustrated in the results include sprung mass vertical displacement and acceleration and pitch and roll angles For the unsprung masses vertical displacement of the wheel's center of gravity, tyre force and suspension travel are included As an aid to the analysis, the road profile is superposed in some of the simulation results

3.1.2 Steady-State Manoeuvres

The primary task of the steady-state manoeuvre simulations is to obtain static and/or steady-state response information in order to assess the linearity of the model for the various quantities of motion involved in response to the already mentioned inputs and disturbances. Another important purpose was to determine directional stability parameters, related to the specific vehicle being modelled, such as static margin, understeer or oversteer behaviour, etc. Simulation runs which are executed include cornering with constant speed and different steering angles, cornering with constant steering angle and various forward velocities and transversing sinusoidal road profiles of different heights and lengths.

In the cornering at different speed simulation the vehicle is considered to be driving at various constant speeds when the vehicle is steered by a parabola to step steering input. It is intended to cover a realistic range of operating conditions of the vehicle.

The variables which are analysed as functions of forward velocity include pitch and roll angles, lateral velocity, yaw rate, sideslip angle and lateral acceleration of the sprung mass and slip angle of the wheels.

The cornering at different steering angle simulation is similar to the one previously described, but in this case the forward velocity is kept constant and larger values of lateral acceleration are obtained by performing the cornering manoeuvres at different values of steering angles.

In order to investigate the nonlinearities of the quantities of motion in relation to road disturbances, their behaviour are analysed for the vehicle driving over a sinusoidal road profile with various amplitudes.

3.2 Analysis of Simulation Results

It has been a traditional practice in vehicle dynamics studies [3, 6, 11] to divide the quantities of motion in groups according to which aspects of vehicle motion they are related to. As a result of this, the areas of performance, handling and ride studies have been established. Although the purpose of the present work is to look at the automotive vehicle as a total system, these divisions are well consolidated by their use in practice and also they can be very helpful in understanding overall vehicle behaviour and the relationship amongst the innumerable variables of motion.

The area of performance studies or longitudinal dynamics, is concerned with the behaviour of the vehicle when driving straight ahead or at very small lateral accelerations. Vehicle handling or stability refers to a vehicle's behaviour in response to the application of lateral forces and yawing moments caused by steering inputs and vehicle motion. It is also concerned with vehicle lateral stability when driving straight ahead. Ride studies are concerned with a a vehicle s behaviour as it transverses uneven road surfaces. Under certain operating conditions, these areas can be studied separately, as has often been the case. However the coupling amongst these variables must be taken into consideration under other conditions, especially limit manoeuvres. The objectives of this simulation analysis is to look at these couplings, their magnitude and nature.

3.2.1 Definition of a Driving Envelope

The concept of a flying envelope has been originated in aeronautical research and it refers to the operating range of an airplane in terms of its speed normally expressed in mach number, and the altitude range which it covers. A similar concept called driving range has also been used in automotive engineering [6] and it usually refers to the range of longitudinal and lateral accelerations and forward velocity in which a car can operate, given a certain road condition.

In the present case, the concept of driving envelope are used in a different manner. The idea is to define an operating range, or set of operating ranges, in which a linearised, or set of interrelated linearised conditions, would still be valid and therefore amenable for control design and analysis purposes. This analysis is based on the previous simulation results.

The quantities which are used to define the envelope are longitudinal, lateral and vertical quantities. Its concept is diagrammatically illustrated in figure 5. The figure represents the limits of linear validity of the models in the X, Y and Z directions.

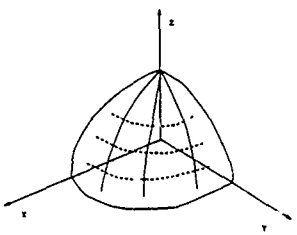


Figure 5. Concept of Driving Envelope

The transient manoeuvre simulations are analysed in comparison to the expected behaviour of a linear system resulting from an eigenvalue-eigenvector analysis around the corresponding operating points. Also, in order to allow a comparison between the various linear system representation, that is, eigenvalues, eigenvectors and system matrices, around different operating points, a certain driving condition will be taken as a *reference* configuration.

The steady-state manoeuvres will be used to obtain the static and/or gain relationship between the various variables of motion and the inputs or disturbances. In this case it will be possible to analyse the strength of the coupling between these variables, and how linear this coupling is. This analysis, together with the previous one, allow the decision of a range of operation in which a linear representation of the system is still valid, and further studies, such as model order reduction, controller analysis and design for active systems, etc. is possible.

3.2.2 Eigenvalue Analysis (transient)

The analysis of the transient motion of the previous simulation runs is performed based on an eigenvalue/eigenvector analysis of the matrices of the linearised system, determined at certain operating conditions. As described earlier, motion quantities are divided according to which modes they are associated with and to which they contribute more significantly. In this sense, some quantities of motion are either taken into consideration or neglected, according to the circumstances of the simulation run.

Steering inputs are related to a vehicle's handling behaviour which is concerned with the ability of the vehicle to change directions at the driver's request and to maintain directional stability at various operating conditions.

The variables of motion which seem to undergo significant variation during cornering manoeuvres include sprung mass yaw rate, lateral velocity or sideslip angle, lateral acceleration and roll angle, and unsprung masses slip angles. Related quantities which are important for the motion, and/or stability purposes, are the tyre forces. Figure 6 depicts some variables of a typical test.

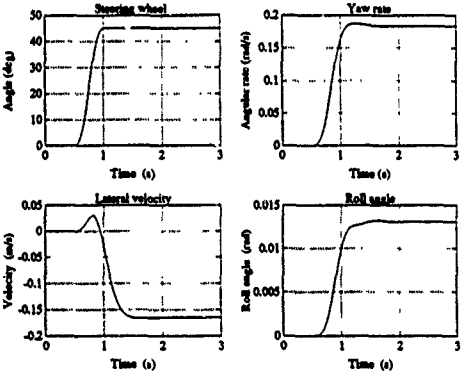


Figure 6: Response of the sprung mass to a steering input

Vehicle directional stability characteristics are related to the eigenvalues of the characteristic equation which is associated with the handling modes. However, a number of parameters which specify a vehicle's handling behaviour, and therefore stability, have been derived from simple models and are well established and understood in the vehicle dynamics community. For handling studies, three *parameters* play a central role in defining vehicle's behaviour:

- front cornering stiffness (gradient);

	Linear		Nonlinear	
Speed	ω_n	ξ	ω_n	ξ
m/s	rd/s	-	rd/s	-
5.0	31.11	1.0	28.96	1.0
	22.57	1 0	23.62	1.0
10.0	13.99	0 97	13.72	0.96
15.0	9.99	0.90	9.72	0.91
20.0	8.13	0.83	7.85	0.85
40.0	5.80	0 59	5.47	0.61
100.0	5.01	0.29	4.92	0.28

Table 5: Handling mode frequencies and damping factors as function of forward velocity for 2 tyre models

- rear cornering stiffness (gradient) and
- location of the vehicle's centre of gravity in relation to the wheelbase.

The value of these parameters determine a vehicle's transient and steady-state characteristics, as well as directional stability properties The handling mode characteristics of a vehicle are shown to be those most susceptible to variations in operating conditions. The *quantity of motion* which seems to affect the handling characteristics most significantly is the vehicle's *forward velocity*. The eigenvalue analysis of the Jacobian of the nonlinear vehicle model shows that for the vehicle driving in a straight line, the handling modes, yaw rate and lateral velocity (or sideslip angle, because forward velocity is assumed to be constant) show the following behaviour with increased speed:

- decrease in natural frequency;
- decrease in damping factor.

This situation is illustrated in table 5.

Because the eigenanalysis at these operating points results in the full set of frequencies, damping factors and modes, The effects of steering angle and forward velocity in the ride and performance modes is also obtained. It could be observed that they have not been affected significantly by forward velocity nor steering angle for the present linear tyre model. The limiting situation for using the linearised model is loss of adhesion which happens due to load transfer effects. The variable of motion which is related to load transfer effect and therefore should be used for a control strategy is *lateral acceleration*.

Traction and braking inputs are associated with a vehicle's performance. In the case of the present model, studies are affected and limited due to the lack of engine, transmission and braking system models, since the dynamics of these subsystems also determine vehicle longitudinal motion. However, it could be observed that ride modes are exercised by performance manoeuvres and that the

variation of wheel loads affect vehicle lateral stability in combined manoeuvres Figures 7 and 8 represents some variables of motion for such simulation.

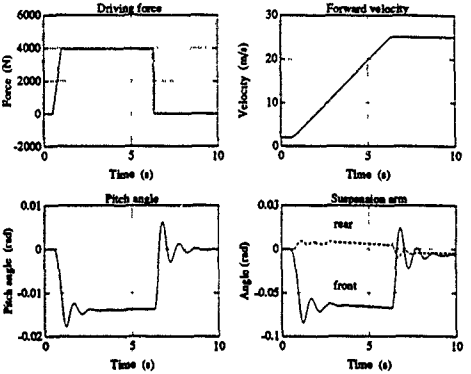


Figure 7: Response of the sprung mass to a acceleration force

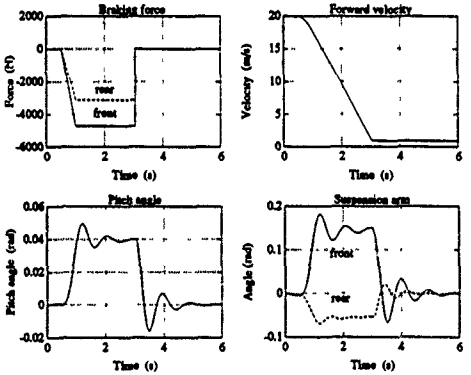


Figure 8: Response of the sprung mass to a deceleration force

Road vertical disturbance simulations are aimed at testing vehicle ride modes The study of vehicle ride modes assumes a contact point model for the interaction

between tyre and road in the vertical direction. This situation is equivalent to a lumped parameter model for the tyre vertical force characteristics. One such test result are illustrated in figures 9 and 10.

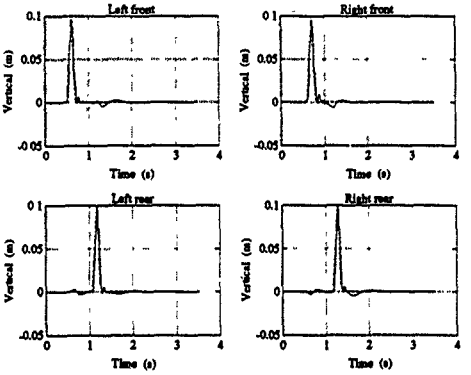


Figure 9: Response of the wheels to a road input

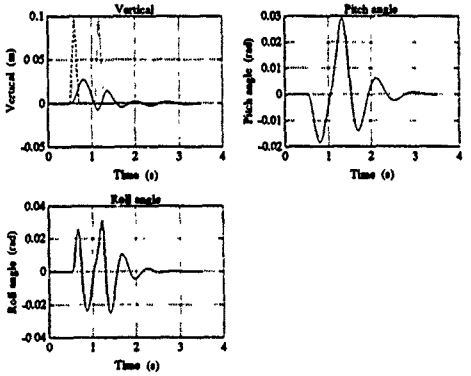


Figure 10: Response of the sprung mass to a road input

Analysing the influence of bump height for the same speed, it can be observed

that, for all variables of motion, the changes in transient behaviour is completely negligible. One cannot even distinguish to which height the simulation refers to, if the scale of the y-axis would not be shown [9].

The behaviour of the sprung mass for the same bump height and for three different speeds shows that, irrespective of the value of the speed, their transient motion *after* the inputs have been applied, in the range in which they get excited by the bump input, are nearly the same, in terms of the frequency of oscillation and damping. In this way, vertical displacement and acceleration and pitch and roll angles show similar behaviour for all three speeds for each one of the bump heights.

These results are interesting in the sense that they confirm that the nonlinear vehicle model does not present significant differences in its transient behaviour when under ride mode types of excitation, in terms of *forward speed*. Together with the previous conclusion concerning the effect of bump height and constant speed, these results provide good insight into the use of the linear model approximation of the nonlinear vehicle model for the area of ride performance. It is worth noting that this area's main concern, from a control point of view, is analysis and design of controlled suspension.

3.2.3 Gain Analysis (steady-state)

The gain, or steady-state analysis, is intended to assess the level of coupling amongst the variables of motion, the amount of nonlinearities in these couplings and also to determine vehicle directional stability parameters.

For this analysis, three types of steady-state simulations have been performed:

- cornering at constant speed with various steering angles;
- cornering with constant steering angle at various speeds;
- crossing of obstacles with sinusoidal profile of different heights, and at different speeds.

The results presented in the table 6 are pitch and roll angles, yaw rate, lateral velocity and acceleration, sideslip angle and slip angle of the wheels. They are calculated as a function of the steering angle for the constant speed manoeuvre. For the constant steering manoeuvres the same quantities as above are calculated in terms of forward velocity.

As it can be deduced from this table, all variables chosen to be retained showed to be affected by the steering input. The pitch angle showed the smallest influence of all. This result confirms the widespread practice to ignore vehicle pitch from the state variables in handling studies, and when it is considered, its effect is calculated in an approximated manner.

It also confirms real vehicle behaviour, because a vehicle does not suffer large pitch angle variations in any normal operating condition. The other variables chosen for the sprung mass movement, yaw rate (u_4), and lateral velocity in vehicle body (V_y), and slip angle (α_i) for the wheels, are the main quantities in handling studies and their importance for handling studies is obvious.

Constant Speed Input: Steering (rd)			Constant Steering Input: Speed (m/s)		
Output	Gain	Corr.	Out.	Gain	Corr.
$a_y\ (m/s^2)$	4.5921	1.0	a_y	0.089	1.0
$q_5\ (rd)$	0.0004	0.98	q_5	0.002e-3	0.79
$q_6\ (rd)$	0.0165	1.0	q_6	0.0003	0.99
$u_4\ (rd/s)$	0.2304	1.0	u_4	0.0013	0.79
$V_y\ (m/s)$	-0.2003	-1.0	V_y	-0.0201	-0.96
$\beta\ (rd/s)$	-0.0100	-1.0	β	-0.0007	-0.99
$\alpha_f\ (rd/s)$	-0.0447	-1.0	α_f	-0.0009	-0.99
$\alpha_r\ (rd/s)$	-0.0277	-1.0	α_r	-0.0005	-0.99

Table 6: Steady-state Gains

For the remaining variables of motion related to vehicle handling, it can be seen that the model coupling concerning forward velocity, can be adequately described by linear relations for vehicle steady-state behaviour. This can have interesting design implications from a control point of view, depending which strategy is adopted to implement the control. If forward velocity is adopted as a system *parameter*, the fact that model characteristics change in a linear way for the steady-state behaviour, together with the conclusions already made about the transient behaviour, may allow the implementation of some kind of adaptive control, for example, as a function of vehicle speed.

The sinusoidal profile obstacle crossing manoeuvres are discussed next. The objectives of these simulations are to verify the linearity of the ride modes of the vehicle model. From a large number of simulation generated frequency responses, it could be observed that there is no significant influence of the input signal amplitude in the vehicle vertical modes, indicating that linear approximation are adequate for these modes as indicated by figure 11.

It can be concluded that the present vehicle model, with linear relations for the spring and damper of the suspension, and for a linear vertical tyre behaviour, as well, can be adequately represented by a linearised model, concerning ride modes behaviour. The limiting factor in this case being the loss of adhesion or compression of the tyre beyond the assumed linear range. These results show that the inertia couplings, in this model, for these modes, are weak.

3.2.4 Determination of a Driving Envelope for Linear Control Analysis and Design

The discussion about a driving envelope for linear control analysis and design is centred at the previous simulation runs.

The areas of vehicle behaviour which are discussed are the same according to the division established for the simulation runs. The effects and consequences of considering them together is also attempted to be discussed.

For the vehicle ride behaviour, with its present tyre and suspension models

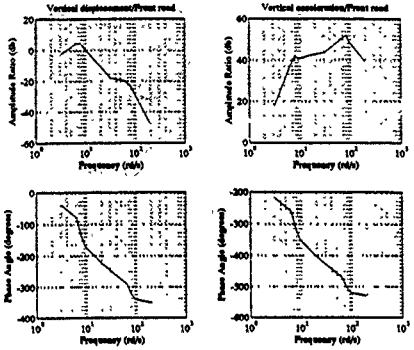


Figure 11: Frequency Response: Amplitudes 0.05, 0.01 and 0.0125 m

the range of validity of linear behaviour seems to be limited only by the loss of contact of the wheels with the road. These conclusions are based on the results of the transient and the steady-state analysis of the previous simulations. For the transient results it can be observed that there is no significant changes in vehicle transient behaviour as the levels of model inputs are varied.

Concerning vehicle performance, not many conclusions can be drawn due to the limitations of the present model powertrain and braking system dynamics, as well as the longitudinal tyre model and which play an important role in vehicle performance mode and its coupling to other vehicle modes, most notoriously to vehicle handling.

Finally, with respect to the handling modes, it seems that for a linear tyre model, quite adequate linear approximations can be made especially for low value of lateral acceleration, based on the analysis of the present simulation results.

In any case, linear model which are adaptive with vehicle speed seem to be recommendable, to successfully implement such controls. The coupling with the ride modes have been previously described and it does not seem likely that a steering control would affect suspension behaviour, but the contrary is true

4 Conclusions

This paper has discussed the importance of the derivation of vehicle chassis models encompassing all aspects of vehicle motion in order to allow the development of integrated vehicle control. It has proposed an approach to modelling vehicle suspension which is based on simple experimental data and uses concepts of differential geometry. The derivation, simulation and analysis of the model was performed using the computer programs SD/FAST, ACSL and MATLAB.

respectively. The model has been validated and therefore it could be used for simulation analysis aimed at obtaining operating ranges in which linear approximations woould be valid. For the present model it can be concluded that the performance and ride modes are not affected significantly by variation in operating conditions nor the steady-state response by the magnitude of the inputs The limiting factors for these aspects of motion are loss of tyre contact. the handling modes are those most affected by changes in operating conditions and the variable of motion which seems to have the largest influence is vehicle speed. However it worth noting that this change do not affect other modes and the coupling occur at the tyre model more than at the vehicle body model

References

[1] *ACSL - Advanced Continuous Simulation Language. Reference Manual.* Mitchell and Gautier Associates, 1987

[2] Costa, A., *Application of Multibody System (MBS) Modelling Techniques to Automotive Vehicle Chassis Simulation for Motion Control Studies* Ph.D. Thesis, University of Warwick, Conventry, England, 1992.

[3] Ellis, J.R., *Vehicle Dynamics*, London Business Books ltd., London, 1969.

[4] Kane, T.R., Levinson, D.A., *Dynamics: Theory and Applications*, McGraw-Hill, New York, 1985.

[5] Kortum, W., Sharp, R., *A Report on the State-of-Affairs on "Application of Computer Codes to Vehicle System Dynamics"*, Vehicle System Dynamics, vol. 20, no. 3-4, pp 177-184, 1991.

[6] Lugner, P., *Horizontal Motion of Automobiles. Theoretical and Practical Investigations*, Dynamics of High-Speed Vehicles, CISM Course no. 274, Springer-Verlag, 1982.

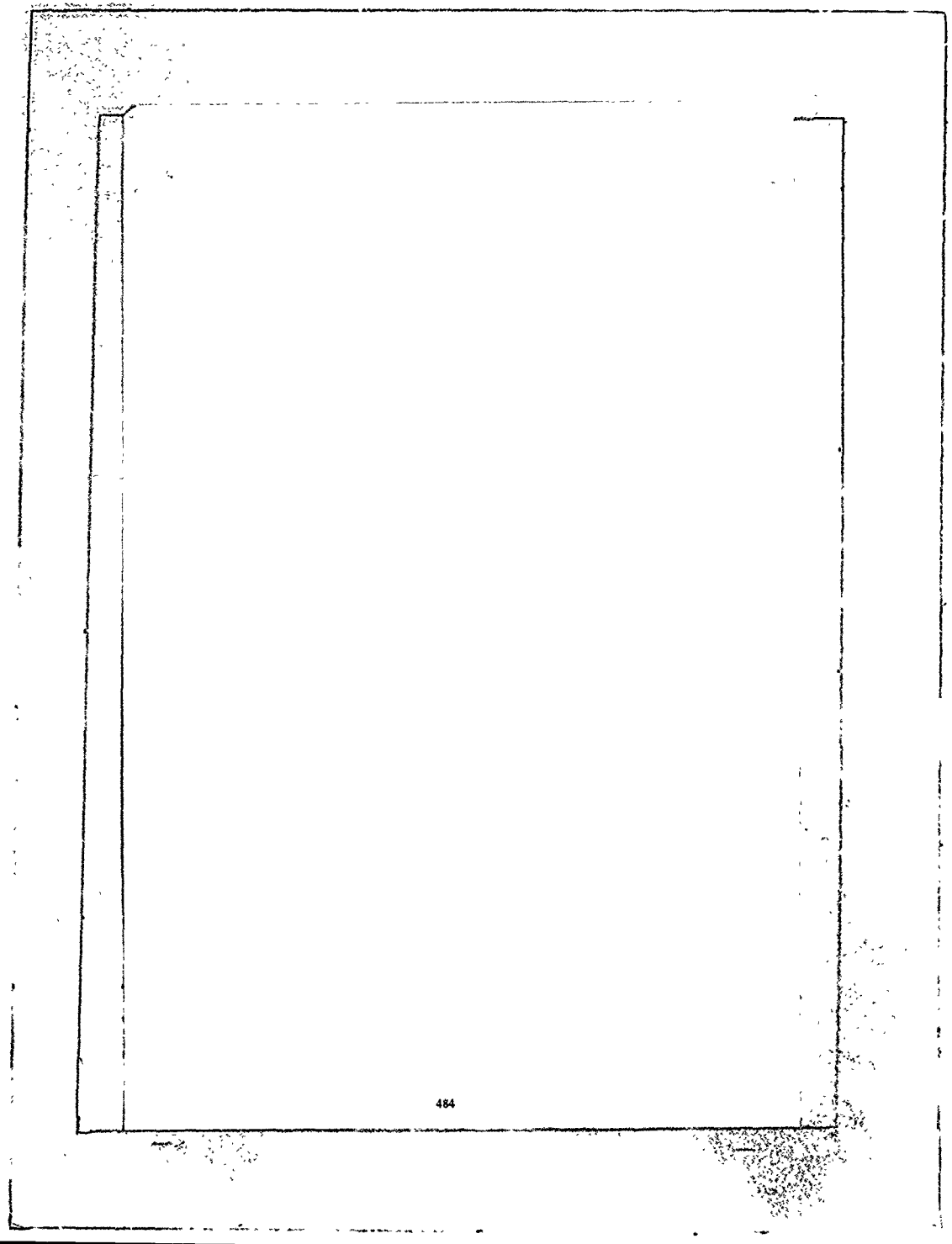
[7] *Pro-Matlab: User's Guide - For Sun Workstations*, The MathWorks Inc., 1987.

[8] Rosenthal, D.E., Sherman, M A., *High Performance Multibody Simulations via Symbolic Equation Manipulation and Kane's Method*, J. Astr. Sci., 34, 223-239, 1986

[9] Schiehlen, W.O., *Introduction to Vehicle Dynamics*, Dynamics of High-Speed Vehicles, CISM Course no. 274, Springer Verlag, 1982.

[10] *SD/FAST User's Manual - version b1.1 pre-release edition*, 1990

[11] Segel, L., *Theoretical Prediction and Experimental Substantiation of The Response of the Automobile to Steering Control.* IMech.E, A.D , London, 1956.



NUMERICAL INVESTIGATION OF THE INFLUENCE OF THE SHOCK ABSORBER ON THE VERTICAL FORCE TRANSMISSIBILITY OF A MCPHERSON SUSPENSION

E. Pisino, J. Giacomin, P. Campanile

Centro Ricerche Fiat
Strada Torino 50
10043 Orbassano (TO)
Italia

Abstract

This paper discusses a numerical investigation regarding the effect of the shock absorber damping curve on the dynamic behavior of a vehicle McPherson suspension. The study focuses on the suspension behavior in the vertical direction. Both single harmonic and pseudo-random waveforms were used for the input force. The vertical force transmissibility was calculated from the wheel hub to the McPherson dome for various levels of the forcing function. The results provide a description of the behavior of the suspension unit as a function of the characteristics of both the damper and the force input to the system.

1. Introduction

Automobile suspension units include a number of elements which possess nonlinear characteristics in either displacement or velocity. It has been noted that correct modeling of these nonlinear characteristics is indispensable [1] when studying the dynamic behavior of a suspension unit, and hence its contribution to overall vehicle vibrational comfort. It has also been shown [2] that the most important nonlinearity is that of the shock absorber damping characteristic. The importance of the damper has stimulated a number of studies regarding the modeling and specification of shock absorbers [3-9].

The objective of the present study is to quantify the influence of the damper curve on the vertical force transmissibility from the wheel hub to the upper dome. The upper McPherson dome was chosen as the output measurement point because it is the location at which the greatest static and dynamic forces enter the car body from the suspension unit. The vertical direction has been isolated for the purposes of this study because it is the principal vibrational direction for an automobile on a typical road.

2. The model

Figure 1 presents a McPherson front suspension as found on small to medium European automobiles. Figure 2 presents the model developed for this study which consists of seven node points as defined in Table 1. Three of the nodes are centers of mass. The first center of mass is that of the lower arm (m_5), the second is that of the hub/wheel/strut assembly (m_3), while the third center of mass is the upper half of the strut assembly (m_2). The mass and inertia values are listed in Table 2.

Other elements of the model are the linear main spring (ks), the damper (c) and the linear translational (K_{1x}, K_{1y}, K_{1z}, K_{6x}, K_{6y}, K_{6z}, K_{7x}, K_{7y}, K_{7z}) and torsional (T_{1x}, T_{1y}, T_{1z}, T_{6x}, T_{6y}, T_{6z}, T_{7x}, T_{7y}, T_{7z}) stiffness parameters of the rubber bushings that attach the suspension to the car body. These stiffness values are listed in Table 3. The model also includes a spherical joint, a slider joint and two cylindrical joints which apply the kinematic constraints of the suspension design. The end of travel snubbers were not modeled in this study.

The force input to the model was applied at the hub (m_3) in the vertical direction. The force output was measured at the McPherson dome in the vertical direction, which corresponds to the force transmitted across the vertical stiffness K_{1z}. The suspension unit is not attached to the vehicle in the present study but rather to ground (Figure 2). The model simulates a bench test in which the suspension attachment points are mounted to a rigid frame.

Figures 3 and 4 present the three damping curves considered. The damper of Figure 3 is representative of a medium European automobile and will hence be referred to as normal production. The damper setting is highly asymmetric, producing most of the damping force in the rebound stroke (positive velocity). This damper curve can be divided into two regions indicated as I and II. If we define the viscous damping coefficient as

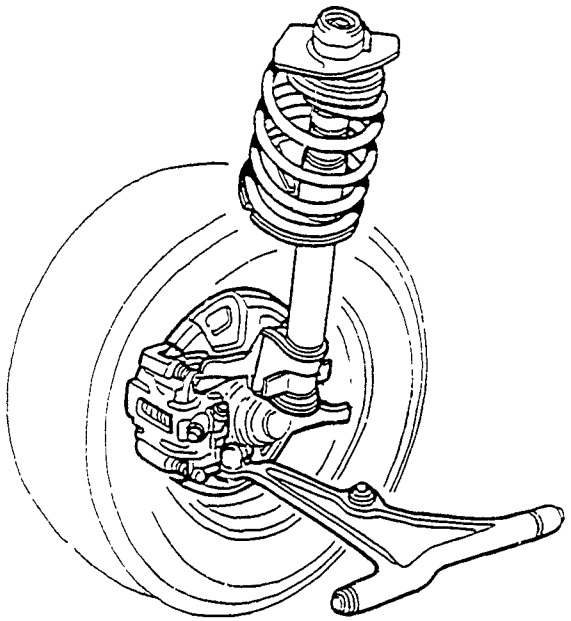


Figure 1) Typical McPherson front suspension.

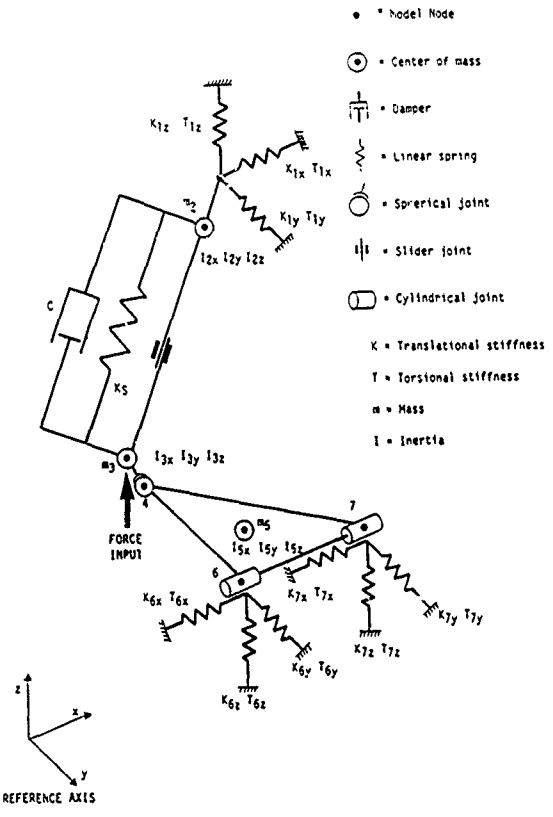


Figure 2) McPherson front suspension model.

Table 1) Node points of the suspension model.

NODE	x, y AND z COORDINATES [mm]
1	(30.6, -586, 612)
2	(16, -592, 333.5)
3	(0, -747, 55)
4	(-4.3, -708, -53.5)
5	(100, -500, 50)
6	(-10, -398, -38)
7	(302. -398. -33 5)

Table 2) Mass and inertia data of the suspension model.

MASS or INERTIA	VALUE [Kg] ; [Kg * mm² mm]
m2	1
m3	38.8
m5	3.5
(I2x, I2y, I2z)	(1, 1, 1)
(I3x, I3y, I3z)	(650E3, 110E4, 550E3)
(I5x, I5y, I5z)	(14E3, 27.3E3, 41 3E3)

Table 3) Translational and torsional stiffness values of the suspension model.

TRANSLATIONAL or TORSIONAL STIFFNESS	VALUE [N/mm] ; [N * mm/rad]
Ks	15
(K1x, K1y, K1z)	(1786, 1786, 570)
(T1x, T1y, T1z)	(10000, 10000, 10000)
(K6x, K6y, K6z)	(312, 4458, 4458)
(T6x, T6y, T6z)	(10000, 10000, 10000)
(K7x, K7y, K7z)	(588, 1214, 1214)
(T7x, T7y, T7z)	(10000, 10000, 10000)

$$c = \frac{df}{dv} \qquad (1)$$

where f is the instantaneous force and v the instantaneous velocity of the absorber, we can describe the curve used in this study as having a varying damping coefficient in region I but constant damping coefficients in region II.

The second damping curve studied (Figure 4) was linear, and was derived from the normal production curve. A linear least squares interpolation was performed to obtain a constant damping coefficient equal to the average of the normal production damper over the velocity range of interest. A correction (shift) was performed so as to produce zero force at the origin.

The third damping curve (Figure 4) is bilinear. It is obtained by rotating the two halves of the viscous damper by a constant but opposite angle. This rotation maintains the same area under the damping curve, but introduces a discontinuity at the origin.

The three damping curves chosen are different in nature, but remove similar amounts of kinetic energy in the velocity interval of this study (0 to 1.2 m/s).

3. Inputs studied

Two types of force input were chosen. The first was single harmonic input (sine) and the second was a band-limited pseudo-random signal (Figure 5). The frequency range of study was from 0 to 50 Hz. The force levels chosen were representative of typical operating loads.

Harmonic input was chosen for two reasons. The first is that a single steady state harmonic oscillation permits a simpler evaluation of system nonlinearities. There are many studies in the related literature which utilize this approach [10-14]. The second reason for choosing a harmonic input is that there exist road inputs such as long road undulations which are similar in nature.

Eight seconds of time domain integration were performed with each sinusoidal force input in order to obtain the system response. The integration time step utilized was 512 points per second. One hundred frequencies were evaluated in the range from 1 to 50 Hz. Five different peak force amplitudes were also utilized. For the three damper characteristics studied this gives a total of 1500 integration runs.

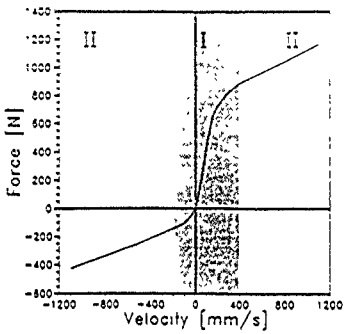


Figure 3) Damping curve of the normal production absorber.

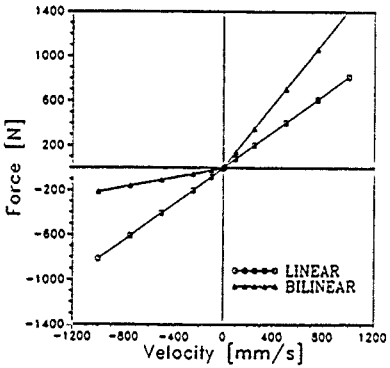


Figure 4) Damping curves for the cases of linear and bilinear damping.

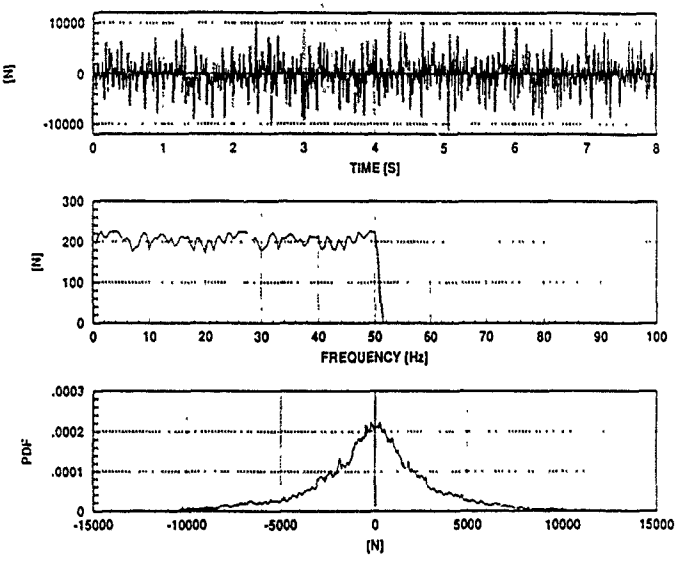


Figure 5) Band-limited pseudo-random force input.

The pseudo-random force input was chosen so as to have a band-limited alternative to single harmonic input. Most road inputs have significant content over a wide frequency band. The frequency content of this input covered the range from 0 to 50 Hz, with one frequency line every .125 Hz, and randomized phases. Seven different RMS levels were created by first generating a base signal, then multiplying it by six constant values. This had the effect of changing the RMS level while maintaining the crest factor (3.95) and phase relationships unaltered. An example of pseudo-random force input is presented in Figure 5.

As in the case of single harmonic input, eight seconds of time domain integration were performed with a time step of 512 points per second. Seven different RMS force levels were utilized. For the three dampers studied this gave a total of 21 integration runs.

4. Results

The model of Figure 2 was assembled and solved with the ADAMS (Appendix 1) program for the various inputs studied. The behavior of the suspension unit was studied by calculating the gain of the force transmissibility in the vertical direction from the hub (m_3) to the McPherson dome (K_{1z} stiffness). The gain was defined in two ways. For the case of single harmonic force input the gain was defined as

$$\text{Gain} = \left| \frac{y}{x} \right| \qquad (2)$$

where: y = maximum output force
x = maximum input force

For the case of the band-limited pseudo-random force input the gain was chosen as

$$\text{Gain} = \left| H_1(\omega) \right| = \frac{\left| Sxy \right|}{Sxx} \qquad (3)$$

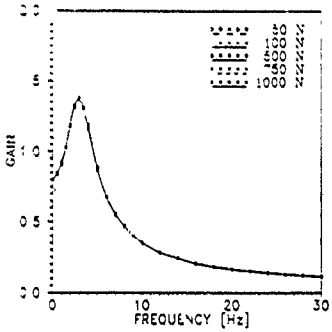
where: Sxy = cross power spectral density
Sxx = input power spectral density

Of the possible estimates for the frequency response function, the H_1 estimator was chosen because it produces the best unbiased linear model of the system [15] in absence of noise on the input. This characteristic is appropriate in the present study because the input force is specified, with noise due to energy transfer present only on the output.

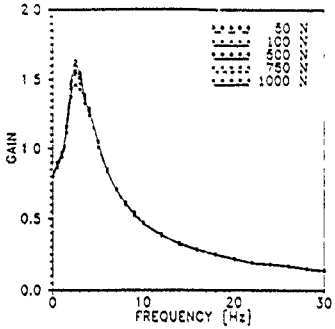
Figure 6 presents the force gain of the suspension in the vertical direction for each of the three dampers when subjected to single harmonic force inputs of different peak amplitude level. The results obtained with the linear damper indicate that the entire model is basically linear. The geometrical nonlinearities were found to be small for the suspension rotations induced (±3 degrees). The situation changes somewhat in the case of the bilinear damper, where nonlinear effects are now visible in the immediate neighborhood of the main suspension resonance (2.95 Hz).

The most pronounced nonlinear effects are produced, however, by the normal production damper (Figure 6-c). Two basic types of behavior are immediately visible. The first type regards those integration runs in which the achieved velocities of the damper were confined to only the first region of the damping curve, that in which the viscous damping coefficient varies. For region I oscillations, increases of the input force produce an increase in the harmonic content of the transmitted force. The harmonics add to the transmitted force at the driving frequencies thus increasing the system gain as calculated by equation (2). The second type of behavior is produced when the damper works for significant periods of time on both the first (I) and second (II) regions of the damping curve. In this situation the system gain becomes more linear as the input level rises (using more of the region of constant viscous damping ratio).

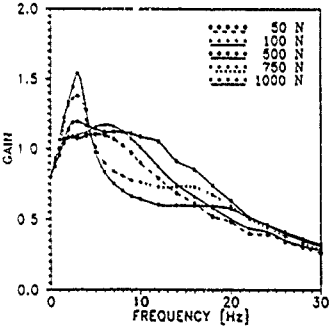
Another representation of the behavior of the suspension unit when subjected to single harmonic input is provided by Figure 7, which presents acceleration response spectra of the hub/wheel/strut assembly when subjected to 3 Hz sinusoidal force input. Spectra are presented for both the bilinear and normal production dampers. The Figure plots the ratio of the acceleration response to the peak force as a function of both frequency and peak force. For the normal production damper, the growth of harmonics with increasing input level is evident from the spectra for the sinusoidal force inputs of 250, 500 and 750 Newtons. The reduced harmonic content of the acceleration response when oscillating in both region I and II is visible in the data relative to the 1000 Newton input force. Two further points



a)



b)

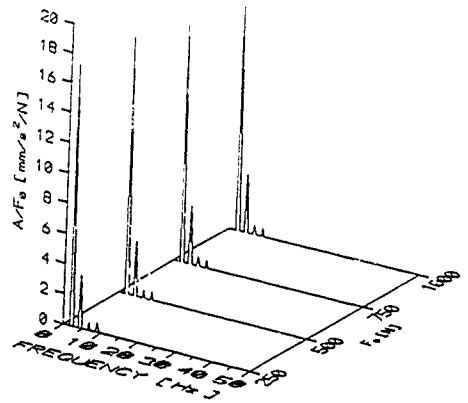


c)

Figure 6) Vertical force transmissibility of the suspension when subjected to single harmonic input.

a) linear damper
b) bilinear damper
c) production damper

a)



b)

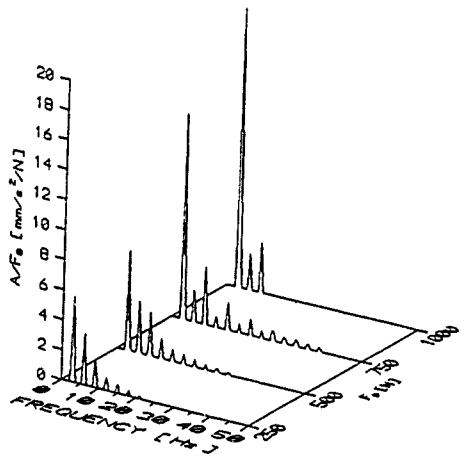
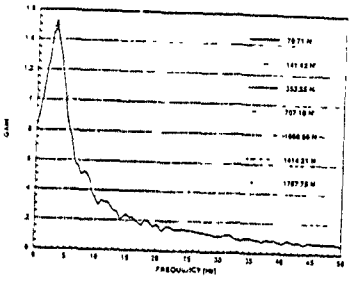
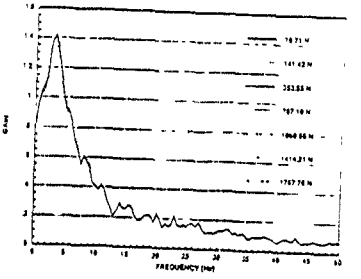


Figure 7) Normalized acceleration spectra of the hub/wheel/strut assembly when subjected to 3 Hz single harmonic input.

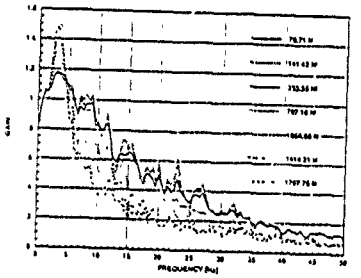
a) bilinear damper
b) production damper



a)



b)



c)

Figure 8) Vertical force transmissibility of the suspension when subjected to pseudo-random input.

a) linear damper
b) bilinear damper
c) production damper

of interest relative to the data of the normal production damper are the reduction in even order harmonics with increasing input force and the reduction in average damping level with increasing amplitude of the input force.

Figure 8 presents the vertical gain of the suspension when exposed to the band-limited pseudo-random force input. Use of this input and of the H1 estimator for the system gain have provided clear evidence of the energy transfer from the system main resonance to both odd and even order harmonics. The harmonics are more strongly excited in the case of the bilinear damper than in the case of the linear damper, where the nonlinearity is only geometric. For the normal production damper, all harmonics of the suspension main resonance are clearly visible in the response up to 9th order. The pseudo-random results confirm the previous observations regarding the two principal regimes of behavior with the normal production damper.

5. Transition points

In the present study, the transition from region I only to region I and II oscillation plays an important role in the behavior of the suspension with the normal production damper. The results indicate that the suspension behaves very nonlinearly when oscillating below the transition points, but becomes more linear as oscillations explore more of region II beyond the transition points.

Figures 9 and 10 present the probability density function (PDF), cumulative density function (CDF) and damping curve for the two types of oscillation identified. Figure 9 is representative of a region I only oscillation while Figure 10 representative of a region I and II oscillation. The second example remains for a significant number of time steps on region II of the damping curve.

The authors propose that the reduced harmonics, and more linear behavior of the suspension with increasing amplitude of the force input is to be attributed to the increasing usage of the linear segments of the damping curve (region II). While region I oscillations are exposed exclusively to a nonlinear damping law, region I and II oscillations are exposed to linear damping for a portion of the cycle that increases as the force amplitude increases.

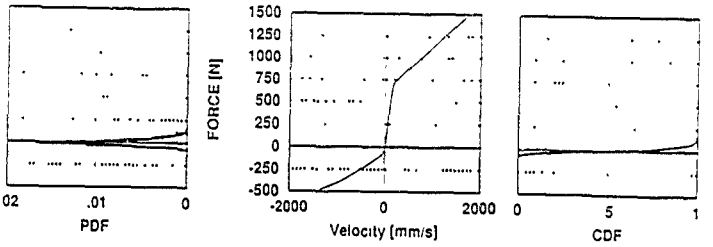


Figure 9) Probability density function, damping curve and cumulative density function for a region I only oscillation.

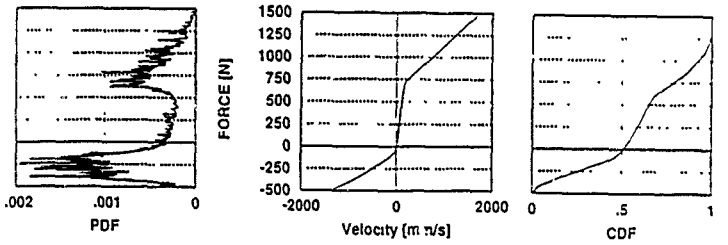


Figure 10) Probability density function, damping curve and cumulative density function for a region I and II oscillation.

6. Conclusions and recommendations

Recent research has indicated that the shock absorber is the major cause of the nonlinear behavior of automobile suspensions in the range from 0 to 50 Hz. The present study has investigated the effect of the damping curve on the vertical force transmitted from the hub to the dome of a McPherson suspension unit.

The vertical force transmissibility from hub to dome for the case of the linear damper has shown that the geometrical nonlinearities are small in the range of input forces studied. The vertical force transmissibility with the bilinear damper was found to be similar to that of the linear damper, but with a stronger harmonic content.

The normal production damper was found to produce a wide range of suspension characteristics depending on the level and type of input. This damper produced highly noticeable harmonics. The results suggest that the location of the transition points from region I to region II of the damping curve plays an important role in determining the overall characteristics of the suspension unit. The results also suggest that the harmonic content of the suspension response is very sensitive to the local variation of the viscous damping coefficient, the strongest harmonics being specifically associated with the regions of highest variation in the viscous damping coefficient.

Areas of future study include: performing a detailed investigation of the energy transfer mechanisms of the current suspension model, investigating the effect of introducing end of travel snubbers to the current suspension model, and adding a tire model and quarter car free mass so as to investigate how the shock absorber influences the force input produced by known road profiles. These activities will further the knowledge of suspension behavior, thus making easier the interpretation of the complex data generated by complete vehicle models and tests.

APPENDIX 1

The object of this Appendix is to describe in very broad terms the operation of the ADAMS program. ADAMS is an environment for building and solving models of multibody (lumped parameter) mechanical systems. An ADAMS model is a data set describing the geometry of a system, the mass and inertia of its components, its stiffness and damping properties, the boundary conditions (constraints) and the system inputs. The data is used to assemble a system of

equations by applying the equations of motion of Lagrange to each mass of the system

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}j}\right] - \frac{\partial L}{\partial qj} + \sum_{i=1}^{m} \frac{\partial \phi i}{\partial qj} \lambda i = Fj \qquad \text{for } j = 1,..n \qquad (A1)$$

$$\phi i = 0 \qquad (A2)$$

where L = the Lagrangian (T-V)
T = kinetic energy
V = potential energy
q = vector of generalized coordinates
F = vector of externally applied forces
ϕ = vector of rigid constraints
λ = vector reaction forces due to the constraints

Equations A1 and A2 describe a constrained equilibrium of the form

$$\begin{bmatrix} \Sigma F \\ \phi \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \qquad (A3)$$

where ΣF and ϕ are nonlinear functions of the system states, the Lagrange multipliers and the time step

$$\Sigma F = f(\ddot{q}, \dot{q}, q, \lambda, t)$$

$$\phi = f(\dot{q}, q, t)$$

The second order terms in equation A3 are reduced to first order by the introduction of a new dependent variable

$$\begin{bmatrix} \Sigma F \\ \dot{q}-u \\ \phi \end{bmatrix} = \begin{bmatrix} 0 \end{bmatrix} \qquad (A4)$$

where u is a vector of generalized velocities. Therefore the sum of forces term of equation A4 becomes

$$\Sigma F = f(\dot{u},\dot{q},q,\lambda,t)$$

The above formulation produces a set of differential and algebraic equations. These equations are solved by utilizing a Gear predictor-corrector algorithm [16-19].

The predictor operations estimate the system response at the next time step (n+1) based on the response at previous points. A polynomial of order k is fitted to the previous values of the system states of each of the generalized coordinates of the system, and the polynomial thus calculated is used to evaluate a truncated Taylor series expansion to obtain the system response at time n+1

$$Y_{n+1} = Y_n + \frac{\partial yn}{\partial t} h + \frac{1}{2!} \frac{\partial^2 yn}{\partial t^2} h^2 \qquad (A5)$$

where y is the system state and h is the integration time step. If the estimate of the system states does not satisfy the equilibrium condition of equation A4 to the desired error tolerance, corrector operations are applied to adjust the state variables.

Corrector operations begin by utilizing the estimate of yn+1 obtained by the predictor to fit a new k order polynomial to the generalized coordinates. This new polynomial is used to evaluate the derivative at time n+1, which is then used to obtain the new estimate

$$Y_{n+1} = -h \beta o \dot{Y}_{n+1} + \sum_{j=1}^{k} a_j Y_{n-j+1} \qquad A6)$$

where $\beta o, a_j$ are the Gear integration coefficients. The above implicit equation is iterated until the solution satisfies the specified error tolerance.
The solution of the system equations provides all the forces, velocities and accelerations that define the boundary equations of the system masses.

REFERENCES

[1] Jolly A., Study of ride comfort using a nonlinear mathematical model of a vehicle suspension. Int. J. of Vehicle Design, Vol 4., No. 3, 233-244 (1983)

[2] Olatunbosun O.A. and Dunn J., An evaluation of the effect of suspension non-linearities on vehicle ride. Second International Conference on Vehicle Comfort, Bologna, Italy, Oct. 14-16 (1992)

[3] Belingardi G. and Campanile P., Improvement of the shock absorber dynamic simulation by the restoring force mapping method. Proc. of the 15th Int. Seminar in Modal Analysis and Structural Dynamics, Leuvan, Belgium, (1990)

[4] Giacomin J., Neural network simulation of an automotive shock absorber. Engng Applic. Artif. Intell., Vol 4., No. 1, 59-64 (1991)

[5] Giacomin J. and Campanile P., Neural network shock absorber model for ADAMS. 8th Tedas NewsConference, Munich, Germany, Oct. 19-21 (1992)

[6] Lang H., A study of the characteristics of automotive hydraulic dampers at high stroking frequencies. P.h.D. Dissertation, The University of Michigan (1977)

[7] Surace C., Worden K. and Tomlinson G.R., On the nonlinear characteristics of automotive shock absorbers. Proc. of the I.Mech.E., Part D - J. of Automobile Engng. (1992)

[8] Surace C., Storer D. and Tomlinson G.R., Characterizing an automotive shock absorber and the dependence on temperature. Proc. of 10th Int. Modal Analysis Conf., San Diego, California (1992)

[9] Worden K. and Tomlinson G.R., Parametric and nonparametric identification of automotive shock absorbers. Proc. of 10th Int. Modal Analysis Conf., San Diego, California (1992)

[10] Shaw S.W. and Holmes P.J., A periodically forced piecewise linear oscillator. J. of Sound and Vib., Vol. 90, No. 1, 129-155 (1983)

[11] Thompson J.M.T , Bokaian A.R. and Ghaffari R., Subharmonic resonances and chaotic motions of a bilinear oscillator. J. Appl. Math., 31, 207-234 (1983)

[12] Thompson J.M.T. and Elvey J.S.N., Elimination of subharmonic resonances of compliant marine structures. Int. J. Mech. Sci., Vol. 26, No. 6-8, 419-426 (1984)

[13] Thompson J.M.T. and Stewart H.B., Nonlinear dynamics and chaos: Geometrical methods for engineers and scientists. John Wiley & Sons Inc., New York (1989)

[14] Natsiavas S. and Babcock C.D., Behavior of unanchored fluid filled tanks subjected to ground excitation. J. Appl. Mech., 55, 654-659 (1988)

[15] Bendat J.S. and Piersol A.G., Random data: Analysis and measurement procedures, 2nd Ed., Wiley-Interscience, New York (1986)

[16] Orlandea N., Node-analogous, sparsity-oriented methods for simulation of mechanical dynamic systems. P.h.D. Dissertation, The University of Michigan (1973)

[17] Mechanical Dynamics, Inc. ADAMS reference guide. Mechanical Dynamics, Inc., Ann Arbor MI (1991)

[18] Gear C.W., The numerical integration of ordinary differential equations. Math Comp. 21, 2, 146-156 (1967)

[19] Gear C.W., Simultaneous numerical solution of differential-algebraic equations. IEEE Trans. on Circuit Theory, Vol. CT-18, No. 1, 89-95 (1971)

IMPACT DYNAMICS OF MULTIBODY SYSTEMS WITH FRICTIONAL CONTACT USING JOINT COORDINATES AND CANONICAL EQUATIONS OF MOTION

M. SEABRA PEREIRA
Departamento de Engenharia Mecânica
Instituto Superior Técnico
1096 Lisboa, Lisboc CODEX
Portugal

P. NIKRAVESH
Dept. of Aerospace and Mechanical Eng
The University of Arizona
Tucson, AZ 85721
USA

ABSTRACT: This paper presents a methodology in computational dynamics for the analysis of mechanical systems that undergo intermittent motion. A canonical form of the equations of motion is derived with a minimal set of coordinates and these equations are used in a procedure for balancing the momenta of the system over the period of impact, calculating the jump in the body momenta, velocities discontinuities and rebounds. The effect of dry friction is discussed and a contact law is proposed. The present formulation is extended to open and closed loop mechanical systems where the jumps in the constraint's momenta are also solved. The application of this methodology is illustrated with the study of impact of open loop and closed loop examples.

1. Introduction

Impact occurs in many mechanical systems, such as crushing and feeding machinery. This multibody systems undergo intermittent motion essentially characterized by discontinuities they encounter when two bodies within these systems collide.

The use of the canonical form of equations of motion, in impact analysis involving impulse and momentum variables, has been presented in several text books[1-3], where solutions are obtained for velocity changes during impact of particles and simple examples of rigid body collisions. The problem is more complex for kinetically constrained mechanical systems, once the change in momenta and velocities are not solely due to the impulsive forces of the colliding bodies but also involve the changes in the reactions of the kinematic constraints.

Piecewise analysis formulations have been developed for the solutions of the rebounds in the intermittent motion of the mechanical systems [4-6]. During the period of the impact, the integration of the equations of motion is halted and a momentum balance analysis is performed to calculate the velocity jumps. Impulsive forces are a byproduct of these calculations. In these methods it is assumed that no significant change in the system configuration occur during the collision time which is considered small compared to a typical time scale of the motion before and after the impact.

Lankarani and Nikravesh [7] solved the direct central friction impact of a mechanical system using a canonical form of the equations of motion expressed in terms of a large set of a Cartesian variables. The work presented here starts from that Cartesian canonical form of the equations of motion, then the equations are converted to a minimal set of equations in terms of a set of joint coordinates and associated generalized momenta.

The resulting formulation together with a proposed contact law corresponding to the general oblique impact problem between two bodies of a multibody system is applied to some examples. The role of the coefficient of restitution and the coefficient of friction is discussed and a methodology is suggested to establish the conditions whether or not sliding or stiction can take place.

2. Equations of motion

The equations of motion can be described in terms of different sets of coordinates. If the number of generalized coordinates is greater than the number of system's degrees of freedom, then algebraic equations are required to show the dependency of the coordinates. One such set of coordinates which leads to defining algebraic constraints for the kinematic joints is the so called absolute cartesian coordinates [8].

Another set of generalized coordinates which can provide a minimal set of equations is known as the joint coordinates. In the following sections, the equations of motion in terms of the joint coordinates are discussed.

2.1. STANDARD FORM

In order to specify the position of a rigid body in a global non-moving XYZ coordinate system, it is sufficient to specify the spatial location of the origin (center of mass) and the angular orientation of a body fixed $\xi\eta\zeta$ coordinate system. For the ith body in a multibody system $\mathbf{q}_i$ denotes a vector of coordinates which contains a vector of translational coordinates $\mathbf{r}_i$ and a set of rotational coordinates. Matrix $\mathbf{A}_i$ represents the rotational transformation of the $\xi_i\eta_i\zeta_i$ axes relative to the XYZ axes. A vector of velocities for body i is defined as $\mathbf{v}_i$, which contains a three vector of translational velocities $\dot{\mathbf{r}}_i$ and a three vector of angular velocities ω_i. A vector of accelerations for this body is denoted by $\dot{\mathbf{v}}_i$ which contains $\ddot{\mathbf{r}}_i$ and $\dot{\omega}_i$. For a multibody system containing b bodies, the vector of coordinates, velocities and accelerations are $\mathbf{q}$, $\mathbf{v}$, and $\dot{\mathbf{v}}_i$, respectively, for body i=1,1...,b. Also a generalized mass matrix is denoted by $\mathcal{M}$ and a vector of generalized forces $\mathcal{F}$ is defined for the multibody system.

The relative configurations of two adjacent bodies can be defined by one or more so-called joint coordinates equal in number to the number of relative degrees of freedom between these bodies. The vector of coordinates for an open loop system is denoted by θ containing all of the joint coordinates and the absolute coordinates of a base body if the base body is not the ground. Therefore, vector θ has a dimension k, equal in number of the degrees of freedom of the system. The vector of joint velocities is defined as $\dot{\theta}$. It can be shown that there is a linear transformation between θ and $\mathbf{v}$ as [9-11].

$$\mathbf{v} = \mathbf{B}\dot{\theta} \qquad (1)$$

where B is a nxk matrix. When the number of selected coordinates is equal to the number of degrees of freedom, the generalized equations of motion for an open loop multibody system can be written as

$$M\ddot{\theta} = f \tag{2}$$

where

$$M = B^T \mathcal{M} B \tag{3}$$

$$f = B^T(g - \mathcal{M}\dot{B}\dot{\theta}) \tag{4}$$

In order to solve these equations, a set of initial conditions must also be defined as

$$\theta(0) = \theta^0, \quad \dot{\theta}(0) = \dot{\theta}^0 \tag{5}$$

Assume that there are one or more closed kinematic loops in a multibody system. To derive the equations of motion for such a system, each closed-loop is cut at one of the kinematic joints in order to obtain an open-loop system without defining any joint coordinates for the cut joints. Therefore, vector of joint coordinates θ has a dimension k greater than the number of degrees of freedom of the closed-loop system. If the cut joints are reassembled, the joint coordinates are no longer independent. Therefore, there exist algebraic constraints between the joint coordinates as [12].

$$\Psi(\theta) = 0 \tag{6}$$

The first and second time derivatives of the constraints are

$$\dot{\Psi} \equiv C\dot{\theta} = 0 \tag{7}$$

$$\ddot{\Psi} \equiv C\ddot{\theta} + \dot{C}\dot{\theta} = 0 \tag{8}$$

where C is the Jacobian matrix of the constraints. Then, the differential equations of motion of Eq. 2 are modified as

$$M\ddot{\theta} - C^T v = f \tag{9}$$

where v is a vector of Lagrange multipliers.

Eqs. 6-9 represent a set of differential algebraic equations for a closed loop system. A set of initial conditions, such as the set given by Eq. 5, but consistent with the constraints of Eqs. 6 and 7, must also be defined. These equations can further be reduced to a minimal set of second order differential equations, equal in number of the degrees of freedom of the system [12].

2.2. CANONICAL FORM

The equations of motion for a multibody system can also be derived in terms of the total momenta of the system. The process of converting the equations of motion described in terms of a large set of absolute accelerations to a canonical form has been shown in [7].

In order to transform the open loop equations of motion of Eq. 2 to the canonical form, a vector of joint momenta $\mathbf{p}$ is defined as

$$M\theta = p \tag{10}$$

where

$$p = f + M\dot{\theta} \tag{11}$$

Eqs 10 and 11 form a simultaneous system of 2xk differential equations of the first order which can be considered as the solution of the Lagrangean problem assuming that θ and p are varied independently of each other [13]. For these equations, an appropriate set of initial conditions must be defined as

$$\theta(0) = \theta^0,\ p(0) = p^0 \tag{12}$$

For closed loop systems a number of prescribed scleronomic conditions may be given by Eq. 6 and a mixed representation corresponding to the general canonic transformation is defined as [13]

$$M\theta - C^T\sigma = p \tag{13}$$

where vector of Lagrange multipliers σ is defined as $\sigma = v$. The time derivative of Eq. 13 yields

$$\dot{p} = f + \dot{M}\dot{\theta} - C^T\sigma \tag{14}$$

Equations 13 and 14, in conjunction with Eqs. 6 and 7, provide the constrained equations of motion in canonical form. A proper set of initial conditions such as the set given by Eq. 12 and consistent with the constraints of Eqs. 6 and 7 are also required.

Numerical solution of the canonical equations of motion, for either an open or a closed loop system, can be obtained by introducing integration arrays as

$$y = \begin{bmatrix}\theta\\ p\end{bmatrix}, \qquad \dot{y} = \begin{bmatrix}\dot{\theta}\\ \dot{p}\end{bmatrix} \tag{15}$$

At every integration time step, $\dot{y}$ array is integrated to obtain y. For open loop systems, vector θ is used to determine the absolute coordinates q, then vector p is used to obtain $\dot{\theta}$ and then v is found from Eq.1. Eq.11 yields $\dot{p}$, which in addition to $\dot{\theta}$ provide all the elements of $\dot{y}$ in order to continue with the integration.

For closed loop systems, vector θ is found from the solution of Eqs. 7 and 9, where σ is also found at the same time. Then Eq. 14 is used to find $\dot{p}$. The rest of the process is the same as that of the open loop systems. A possible problem with this procedure is that due to the accumulation of numerical errors during integration, the position constraints of Eq. 6 may become violated. Different techniques for eliminating the possibility of constraint violation can be found in [8].

In a piecewise impact analysis, it is assumed that the contact between points P_i and P_j lasts for a short period from $t^{(-)}$ to $t^{(+)}$, during which the configuration of the system does not change. As a result, the dynamics of the open loop and closed loop systems are governed under this smoothness assumption by Eqs. 10 and 11 and Eq. 13 and 14, respectively. A contact -impact law involving the relative velocity v_r and the generalized forces f must also be provided.

3. Points of Contact and Relative Velocities

Assume that an impact occurs between bodies i and j of a multibody system. The relative velocity between the points of contact P_i and P_j is written in terms of the absolute velocities of these points as

$$\begin{aligned} v_r &= \dot{r}_i^P - \dot{r}_j^P \\ &= D_i v_i - D_j v_j \\ &= D_{ij} v \end{aligned} \tag{16}$$

where D_{ij} is a 3xn incidence matrix and it is a function of the coordinates of bodies i and j only. Substituting Eq. 1 into Eq. 16 yields

$$v_r = R\dot{\theta} \tag{17}$$

where $R = D_{ij} B$ is a 3xk matrix describing a θ dependent linear mapping of the generalized velocities.

The relative velocity vector v_r is represented by its components relative to some chosen orthonormal base (n, t_1, t_2)

We shall assume that the unit vector n is defined in the normal direction to the contact surfaces and directed toward body i. A schematic representation of the contact between bodies i and j is shown in Fig. 1.

The component of vector v_r along the normal direction is

$$\begin{aligned} v_n &= n^T v_r n \\ &= n^T (R\dot{\theta}) n \\ &= (c_n^T \dot{\theta}) n \end{aligned} \tag{18}$$

where

$$c_n^T = n^T R \tag{19}$$

is a k vector. The tangential component of v_r corresponding to the slip velocity is denoted as v_t and lies in the tangential plane (t_1, t_2), perpendicular to n such that

$$v_r = v_t + v_n \tag{20}$$

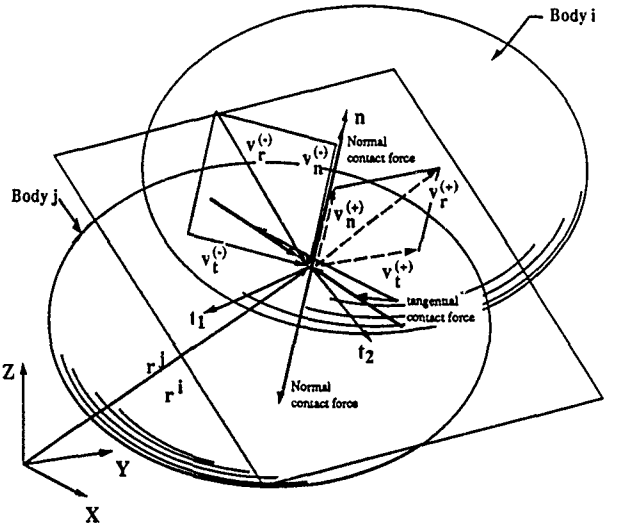


Figure 1. Impact of multibody systems

At $t^{(-)}$ and $t^{(+)}$ the relative velocity vectors are denoted by $v_r^{(-)}$ and $v_r^{(+)}$ respectively. The normal components of these vectors, $v_n^{(-)}$ and $v_n^{(+)}$, are collinear along n. However, their tangential components, $v_t^{(-)}$ and $v_t^{(+)}$, are not necessarily collinear. Velocity jumps are defined as $\Delta v_r = v_r^{(+)} - v_r^{(-)}$, $\Delta v_n = v_n^{(+)} - v_n^{(-)}$ and $\Delta v_t = v_t^{(+)} - v_t^{(-)}$.

A coefficient of restitution in the normal direction is introduced as the ratio between the relative separation velocity and the relative approach velocity as

$$e = -\frac{v_n^{(+)}}{v_n^{(-)}} = -\frac{c_n^T \dot{\theta}^{(+)}}{c_n^T \theta^{(-)}} \qquad (21)$$

The introduction of the intrinsically non-negative value of e has been traditionally associated with the loss of kinetic energy in impacts in which the tangential component of impulse is absent. This occurs when the contacting surfaces are perfectly smooth or in direct/oblique and central/eccentric impacts. The explicit use of e as a kinematic constraint is extended for the analysis of more general impact situations playing a role of

a material constant based on the common assumption that local inelastic material behavior is solely characterized by that coefficient.

From Eq. 20 the relative velocity jump during impact is given by

$$\Delta v_t = \Delta v_t + \Delta v_n \tag{22}$$

Substituting Eq. 17 and Eq. 21 into Eq. 22

$$R\Delta\dot{\theta} = \Delta v_t - n(1+e)c_n^T\dot{\theta}^{(-)} \tag{23}$$

This equation involving the coefficient of restitution, the incoming velocities, the jumps in the generalized velocities and slip velocity will be found to be an important geometric relationship in the piecewise analysis of impact.

4. Impulsive Forces and Impulses

During impact, a pair of impulsive forces, $f^{(i)}$, act at the point of contact between bodies i and j. Superscript (i) indicate the *impulsive* nature of these forces. The corresponding power of the pair of contact forces can be written as

$$\begin{aligned} \mathcal{P} &= v_r^T f^{(i)} \\ &= \theta^T R^T f^{(i)} \\ &= \dot{\theta}^T f^{(i)} \end{aligned} \tag{24}$$

where

$$f^{(i)} = R^T f^{(i)} \tag{25}$$

is the k-dimensional generalized force vector associated with the impulsive force $f^{(i)}$.

The impulsive force $f^{(i)}$ can be decomposed along the normal direction n, and along a tangential direction t in the plane (t_1, t_2).The direction of t will be discussed later on.

Thus

$$f^i = n.f_n^{(i)} + t.f_t^{(i)} \tag{26}$$

and

$$\begin{aligned} f^{(i)} &= R^T.(n.f_n^{(i)} + t.t_t^{(i)}) \\ &= c_n f_n^{(i)} + c_t f_t^{(i)} \end{aligned} \tag{27}$$

where c_t is defined as in Eq. 19 and using t instead of n.

The generalized impulse vector π due to impulsive forces $f^{(i)}$ is

$$\begin{aligned}\pi &= \int_{t(\cdot)}^{t(+)} f^{(i)} dt \\ &= c_n \int_{t(\cdot)}^{t(+)} f_n^{(i)} dt + c_t \int_{t(\cdot)}^{t(+)} f_t^{(i)} dt \\ &= c_n \pi_n + c_t \pi_t\end{aligned} \tag{28}$$

where π_n and π_t are the normal and the tangential impulses due to impulsive forces $f_n^{(i)}$ and $f_t^{(i)}$ acting in the normal and tangential directions, respectively.

The method of obtaining the generalized impulsive forces $f^{(i)}$ and the generalized impulse π through the linear mappings c_n and c_t suggests the usual concept of virtual motions at fixed time and expressed as a family of configurations $\theta(\varepsilon)$ depending on a real variable, ε, in a differentiable manner, with subsequent calculations made at constant

5. Open Loop Systems

From Eq. 10 the generalized velocity jumps can be obtained

$$\Delta\dot{\theta} = M^{-1}\Delta p \tag{29}$$

Integrating the canonical equations of motion described by Eq. 11 for the period of contact, the generalized impulse vector Δp is given as

$$\Delta p = \int_{t(\cdot)}^{t(+)} f \, dt + \int_{t(\cdot)}^{t(+)} \dot{M}\dot{\theta} \, dt \tag{30}$$

The generalized force vector f can be described as the sum of two generalized impulsive and non impulsive forces

$$f = f^{(i)} + f^{(ni)} \tag{31}$$

Since the period of contact is assumed to be very short, i.e., almost zero, only the impulsive forces have nonzero impulse; all other forces are finite including the gyroscopic and Coriollis forces. In fact it can be shown that integrating by parts the last term of Eq. 30

$$\begin{aligned}\int_{t(\cdot)}^{t(+)} \dot{M}\dot{\theta} \, dt &= M\dot{\theta}\Big|_{t(\cdot)}^{t(+)} - M\int_{t(\cdot)}^{t(+)} \ddot{\theta} \, dt \\ &= M\dot{\theta}\Big|_{t(\cdot)}^{t(+)} - M\dot{\theta}\Big|_{t(\cdot)}^{t(+)} = 0\end{aligned} \tag{32}$$

therefore Eq. 30 becomes

$$\Delta p = \int_{t^{(-)}}^{t^{(+)}} f^{(i)} dt = \pi \tag{33}$$

5.1. THE SLIDING PROBLEM FOR OPEN LOOP SYSTEMS

Assuming that, during contact, the process of tangential force generation is that of dry friction, the impulsive force $f^{(i)}$ can be written as

$$f^{(i)} = f_n^i(n + \mu t) \tag{34}$$

where μ represents a friction coefficient. The generalized impulse vector yields for this case

$$\pi = \pi_n(c_n + \mu c_t) \tag{35}$$

Substituting Eqs. 35, 33 and 29 into Eq. 23

$$\mathbf{R}\mathbf{M}^{-1}\pi_n(c_n + \mu c_t) = \Delta v_t - n(1+e)c_n^T\dot{\theta}^{(-)} \tag{36}$$

Premultiply both sides by n^T and realizing that, by definition: Δv_t is perpendicular to n; $n^T n = 1$ and from Eq. 18, the scalar value of v_n is guiven by $v_n^{(-)} = c_n^T\dot{\theta}^{(-)}$, then

$$n^T\mathbf{R}\mathbf{M}^{-1}\pi_n(c_n + \mu c_t) = -(1+e)v_n^{(-)} \tag{37}$$

This scalar equation can be solved for the normal impulse

$$\pi_n = -\frac{(1+e)v_n^{(-)}}{c_n^T\mathbf{M}^{-1}(c_n + \mu c_t)} \tag{38}$$

5.2. THE STICKING PROBLEM FOR OPEN LOOP SYSTEMS

To solve the sticking problem the impulse must be described by Eq. 28 and an extra kinematic condition has to be considered as, in a sticking impact, the outgoing relative tangential velocity vanishes, thus $v_t^{(+)} = 0$ and $\Delta v_t = v_t^{(-)}$. The slip velocity for $t = t^{(-)}$ can be expressed in the more convenient form

$$\begin{aligned} v_t^{(-)} &= v_r^{(-)} - n v_n^{(-)} \\ &= (\mathbf{R} - n.c_n^T)\dot{\theta}^{(-)} \end{aligned} \tag{39}$$

which allow us to write Eq. 23 in the form

$$R\Delta\dot{\theta} = -(R + e\, n.c_n^T)\dot{\theta}^{(-)} \tag{40}$$

In order to write an impulse equation , substitute Eqs. 28, 33 and 29 in Eq. 40 to get

$$RM^{-1}(c_n\pi_n + c_t\pi_t) = -(R + e\, n.c_n^T)\dot{\theta}^{(-)} \tag{41}$$

Finally, premultiplying the preceding equation by n^T and by t^T a set of two momemtum balance impulse equations can be written in matrix form as

$$\begin{bmatrix} c_n^T M^{-1} c_n & c_n^T M^{-1} c_t \\ c_t^T M^{-1} c_n & c_t^T M^{-1} c_t \end{bmatrix} \begin{Bmatrix} \pi_n \\ \pi_t \end{Bmatrix} = \begin{Bmatrix} -(1+e)\, v_n^{(-)} \\ -v_t^{(-)} \end{Bmatrix} \tag{42}$$

Equation 42 is a set of two linear equations in terms of the normal and tangential components of impulse.

6. Closed Loop Systems

The change in the generalized joint velocities during impact must satisfy the constraints of Eq. 7

$$C\Delta\dot{\theta} = 0 \tag{43}$$

From Eq. 13 we have

$$\Delta\dot{\theta} = M^{-1}(\Delta p + C^T\Delta\sigma) \tag{44}$$

Then,

$$CM^{-1}(\Delta p + C^T\Delta\sigma) = 0 \tag{45}$$

Integrating the canonical equations of motion described by Eq. 14 for the period of contact yields

$$\Delta p = \int_{t(-)}^{t(+)} f\, dt + \int_{t(-)}^{t(+)} \dot{M}\dot{\theta}\, dt - \int_{t(-)}^{t(+)} C^T\sigma\, dt \tag{46}$$

All the forces in the right hand side of Eq. 34 are bounded except for impulsive forces. The integral of this bounded forces, including the term containing σ are zero. This can be shown easily by writing

$$\frac{d}{dt}(C^T\sigma) = C^T\dot{\sigma} + C^T\ddot{\sigma} \tag{47}$$

then,

$$\int_{t(-)}^{t(+)} C^T\sigma\, dt = C^T\sigma\Big|_{t(-)}^{t(+)} - C^T\int_{t(-)}^{t(+)} \sigma\, dt$$
$$= C^T\Delta\sigma - C^T\Delta\sigma = 0 \qquad (48)$$

which allow us to conclude that the change in the total momenta for closed loop systems is still a result of the impulsive forces as described by Eq. 33.

6.1 THE SLIDING PROBLEM FOR CLOSED LOOP SYSTEMS

Using the same procedure as in section 5.1 and taking in consideration Eq 44 instead of Eq. 29 an equivalent of Eq. 37 is then obtained for closed loop systems

$$c_n^T M^{-1}[(c_n + \mu c_t)\pi + C^T\Delta\sigma] = -(1+e)c_n^T\dot{\theta}^{(-)} \qquad (50)$$

Equations 46, after substituting Δp from Eq. 33 have the form

$$CM^{-1}(c_n + \mu c_t)\pi + CM^{-1}C^T\Delta\sigma = 0 \qquad (51)$$

which can be appended to Eq. 50 and then written in the matrix form as

$$\begin{bmatrix} CM^{-1}C^T & CM^{-1}(c_n+\mu c_t) \\ c_n^T M^{-1}C^T & c_n^T M^{-1}(c_n+\mu c_t) \end{bmatrix}\begin{Bmatrix}\Delta\sigma \\ \pi\end{Bmatrix} = \begin{Bmatrix} 0 \\ -(1+e)v_n \end{Bmatrix} \qquad (52)$$

which can be solved for the internal impulses $\Delta\sigma$ and the normal component of impulse π acting at the point of contact.

6.2. THE STICKING PROBLEM FOR CLOSED LOOP SYSTEMS

Using the same procedure as in section 5.2 and taking in consideration Eq. 45 instead of Eq. 29 an equivalent of Eq. 42 is then obtained for closed loop systems

$$\begin{bmatrix} CM^{-1}C^T & CM^{-1}c_n & CM^{-1}c_t \\ c_n^T M^{-1}C^T & c_n^T M^{-1}c_n & c_n^T M^{-1}c_t \\ c_t^T M^{-1}C^T & c_t^T M^{-1}c_n & c_t^T M^{-1}c_t \end{bmatrix}\begin{Bmatrix}\Delta\sigma \\ \pi_n \\ \pi_t\end{Bmatrix} = \begin{Bmatrix} 0 \\ -(1+e)v_n \\ -v_t \end{Bmatrix} \qquad (53)$$

which can be solved for the internal impulses $\Delta\sigma$ and the normal and tangential components of impulse π acting at the point of contact.

7. Contact Law

When contact is in effect a contact law is now suggested for the traditional isotropic Coulomb law of friction involving the following:

- $v_t^{(+)}$ is zero if the magnitude of the tangential impulse is less then the μ times the magnitude of the normal impulse.
- when $v_t^{(+)}$ has a nonzero tangential component, $v_t^{(+)} \neq 0$ the tangential impulse has a magnitude μ times the magnitude of the normal impulse.

For planar oblique impacts this contact law involves the following:

$$\text{If } v_t^{(+)} = 0 \text{ then } \pi_t \leq \mu\pi_n \tag{54}$$

$$\text{If } v_t^{(+)} \neq 0 \text{ then } \pi_t = \mu\pi_n \tag{55}$$

and taking, for the moment

$$t = -\frac{v_t^{(-)}}{\left|v_t^{(-)}\right|} \tag{56}$$

which means that the tangential impulse acts along the opposite direction of $v_t^{(-)}$. This is a plausible assumption which will be shown to correspond to a maximum energy loss in planar impacts. However in spatial oblique impacts the tangential velocity generally undergoes a change in direction which enables a direct determination of the direction of the tangential impulse. Formulations using standard Convex Analysis [14-15] have shown that the Coulomb's law of friction is exactly similar to a *law of perfect plasticity* and can be derived from a "principle" of maximal dissipation. As a result the direction of the tangential impulse, t, should be such that the energy loss during impact is maximized.

8. Velocity Jumps and Rebounds

Procedures for updating velocities after the impact are now obtained. Without loss of generality consider the sliding problem for open loop systems described in section 5.1

Taking in consideration Eq. 33 and substituting Eq. 38 in Eq. 29 the jump in the generalized velocities can be obtained

$$\Delta\theta = -M^{-1}(c_n + \mu c_t)\frac{(1+e)v_n^{(-)}}{c_n^T M^{-1}(c_n + \mu c_t)} \tag{57}$$

And from Eqs. 23 and 21 the jump in the tangential velocity can be obtained

$$\Delta v_t = -(R - nc_n^T)M^{-1}(c_n + \mu c_t)\frac{(1+e)v_n^{(-)}}{c_n^T M^{-1}(c_n + \mu c_t)} \tag{58}$$

A procedure for updating velocities in a general oblique impact can be stated for closed (open) loop systems as.

ALGORITHM.

Knowing the positions and velocities at $t^{(-)}$

1. Find π_n and π_t from Eq. 53 (42)
   if $\pi_t \leq \mu\pi_n$ go to 2
   if $\pi_t > \mu\pi_n$ go to 3
2. We have stiction with $v_t^{(+)} = 0$
   2.1 From Eq. 28 evaluate $\Delta p = c_n\pi_n + c_t\pi_t$
   2.2 Go to 4
3. We have sliding
   Find π_n from Eq. 52 (38)
4. Evaluate $\Delta\theta$ from 44 (29)
5. Update joint velocities as $\theta^{(+)} = \theta^{(-)} + \Delta\theta$
6. Resume with integration

9. Examples and Numerical Results

The purpose of this section is to apply the present piecewise analysis to typical examples of mechanical systems and to show the validity of the method by examining the impact responses.

9.1. SLIDER-CRANK MECHANISM

This first example is taken from reference [7] where it was solved using a piecewise formulation with equations of motion described in terms of a large set of absolute cartesian coordinates. This is a closed loop multibody system impacting a sliding body. A schematic representation of the system is shown in figure 2. A set of relative joint coordinates θ_1, θ_2, θ_3 and θ_4 are used to describe the configuration of the system. Due to the closed loop, θ_1, θ_2 and θ_3 must satisfy Eq. 6, and their first time derivatives must satisfy Eq. 43.

The slider crank is driven by a restoring torque such that the crank maintains almost a constant angular velocity. At some instant, the slider (body 3) impacts the free slider (body 4) which is driven inertially to the left at a constant speed. A coefficient of restitution $e = 0.83$ between the blocks is considered.

This case correspond to a central impact problem therefore no tangential relative velocities are observed and no tangential impulsive forces are developed between bodies 3 and 4 during the impact.

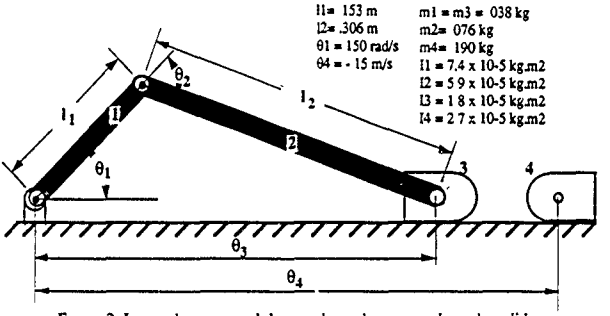


Figure 2. Impact between a slider-crank mechanism and another slider

The results from this analysis and those of reference [7] can be summarized in table 1.

Table 1. Comparison of results

	Present results m/s	Reference [7] m/s
$\Delta\dot{\theta}_4$	41.58	41.4
$\Delta\dot{\theta}_3$	-68.285	-68.3

The implied loss of kinetic energy induced by the impact is given by

$$T_L = T^{(-)} - T^{(+)} = \frac{1}{2}\dot{\theta}^{(-)^T} M\theta^{(-)} - \frac{1}{2}\dot{\theta}^{(+)^T} M\theta^{(+)} \qquad (59)$$

resulting, for central impacts in the form

$$T_L = \frac{1}{2}\frac{1}{c_n^T M^{-1} c_n}(1-e^2)(v_n^{(-)})^2 \qquad (60)$$

For the general oblique impacts the induced energy loss can be shown to be

$$T_L = -\frac{1}{2}\pi_0^2(m_{nn} + 2\mu m_{nt} + \mu^2 m_{tt}) - \pi_n(v_n^{(-)} + \mu v_t^{(-)}) \qquad (61)$$

where

$$m_{nn} = c_n^T M^{-1} c_n \; ; \; m_{nt} = c_n^T M^{-1} c_t \; ; \; m_{tt} = c_t^T M^{-1} c_t \qquad (62)$$

Figure 3 shows a plot of a non dimensionalized form of the energy loss of the system ($\Delta T/T^{(-)}$) for the range of the coefficient of friction $0 \le e \le 1$, and for three different masses of the inertially driven slider (case 1: $m_4 = 0.19$; case 2: $m_4 = 0.95$ and case 3. $m_4 = 1.9$).

For case 1, with e=0, the energy loss is almost equal to the initial kinetic energy of the system before impact, which means that the system came to a halt. For case 2 and 3, where the mass of body 4 was increased, only 54% and 32%, respectively, of the initial kinetic energies are dissipated during perfectly plastic collisions (e = 0) Case 1 is a particular case where the generalized mass of the crank and slider is equivalent to the inertialy driven mass. Case 1 can be reasoned to be "equivalent" to the perfectly plastic central impact of two particles that is known to come to a halt when these masses are equal, thus releasing all kinetic energies.

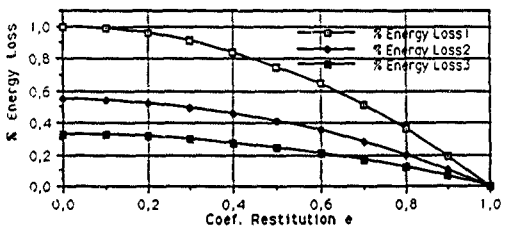


Figure 3. Energy loss versus the coefficient of restitution

9.2. KANE'S DOUBLE PENDULUM - PLANAR CASES

9.2.1. *Case 1.* Fig. 4a) shows two slender rods connected together and to a fixed support with revolute joints, the system is only allowed to move in the XY plane.

The rods are identical and the approaching joint velocities are $\dot{\theta}_1 = -1$ and $\dot{\theta}_2 = -1$ (rad/s). This configuration implies an incoming velocity $v_r^{(-)} = (-2.683t - 1.309n)$.

Figure 5 shows the tangential velocity, $v_t^{(+)}$ for $0 \le e \le 1$.and $-.45 \le \mu \le .45$. Positive values of μ correspond to tangential impulses in the t direction whereas negative values of μ correspond to tangential impulses along the positive X direction. A plot of the energy loss as defined in Eq. 61 is also shown in figure 7.

It can be observed that negative values of μ correspond to smaller energy losses or even energy gains. On the other hand, sticking is not always possible. For example, for values of e<.3, even for large values of μ there is always a positive $v_t^{(+)}$ For larger values of e, sticking can only occur, eventually with energy gains which is not plausible.

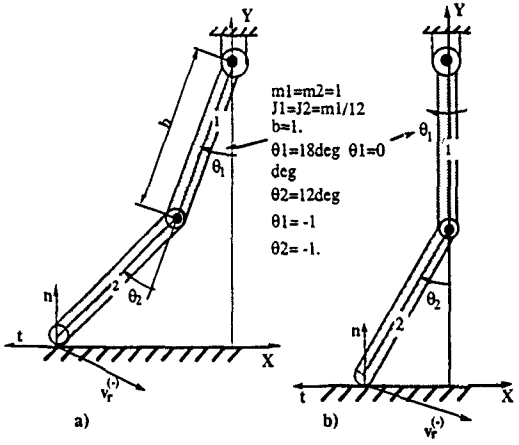


Figure 4. Kane's examples

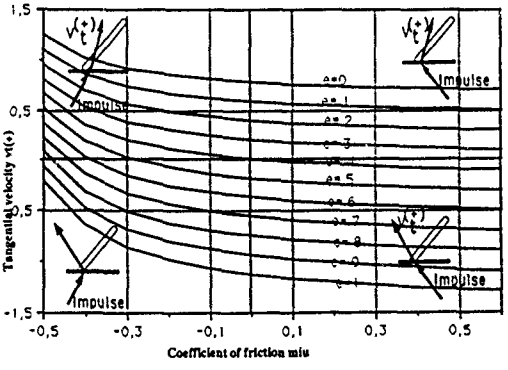


Figure 5. Tangential rebound predictions for the Kane's pendulum a)

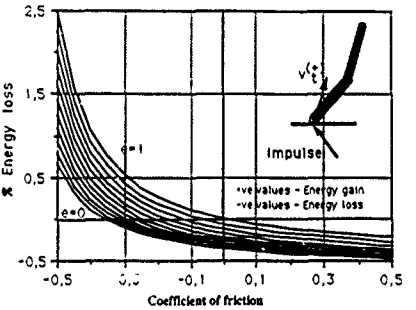
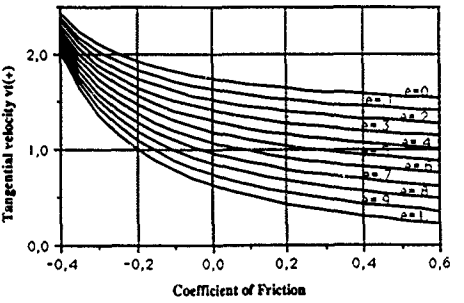


Figure 6. Energy loss for the Kane's pendulum a)

9.2 2. *Case 2.* The double pendulum illustrated in figure 4b) was also anlysed. In this pendulum the first rod is in the vertical position. For this configuration and assuming the same angular velocities as in case a) the relative incoming velocity is now $v_f^{(-)} = (-.707n - 2.844t)$.

Figure 7 shows the predicted outgoing tangential velocities for $0.\leq e \leq 1.$ and $-.45 \leq \mu \leq .45$. The results show that this pendulum never sticks, even for large values of μ the tangential outgoing velocity is always positive.



Fifure 7. Tangential rebound predictions for the Kane's pendulum b)

These results clearly show that the impact conditions are strongly dependent on the mechanical system and its configuration at the time of impact. A method is now proposed to verify analytically whether or not suction can occur.

Without loss of generality consider the open loop case under sticking conditions. As the normal n was defined from body j to body i and the tangential vector t was considered along Δv_t, or, in the opposite sense of $v_t^{(-)}$, thus the solution of Eq. 42 should always yield positive normal and tangential impulses.

According to Crammer's rule

$$\pi_n = \frac{D_1}{D} \quad ; \quad \pi_t = \frac{D_2}{D} \tag{63}$$

where

$$D = \begin{vmatrix} c_n^T M^{-1} c_n & c_n^T M^{-1} c_t \\ c_t^T M^{-1} c_n & c_t^T M^{-1} c_t \end{vmatrix} ; D1 = \begin{vmatrix} -(1+e)\, v_n^{(-)} & c_n^T M^{-1} c_t \\ -v_t^{(-)} & c_t^T M^{-1} c_t \end{vmatrix} ; D_2 = \begin{vmatrix} c_n^T M^{-1} c_n & -(1+e)\, v_n^{(-)} \\ c_t^T M^{-1} c_n & -v_t^{(-)} \end{vmatrix}$$

Therefore if $D > 0$, then $D_1 > 0$ and $D_2 > 0$ or if $D < 0$, then $D_1 < 0$ and $D_2 < 0$. In the present cases $D>0$ then the necessary conditions for suction to occur, can be written as

$$c_t^T M^{-1} c_t v_t^{(-)} - (1+e)\, c_t^T M^{-1} c_t v_n^{(-)} > 0 \tag{64a}$$

$$(1+e) c_t^T M^{-1} c_n\, v_n^{(-)} - c_n^T M^{-1} c_n v_t^{(-)} > 0 \tag{64b}$$

which can be used to define bounds on e for suction to be possible.

9.3. 3D DOUBLE PENDULUM

Consider now that the double pendulum is also rotating around the Y axis with an angular velocity θ_3 as shown in figure 8. As it is shown the vectors t_1 (opposite to the X axis) and t_2 (along the Z axis) define the contact plane. At the time of impact the joint velocities are $\dot{\theta}_1 = -1$ and $\dot{\theta}_2 = -1$ and $\theta_3 = 1$ (rad/s), resulting in the approach velocity $v_t^{(-)} = -2.683\, t_1 + .809\, t_2 - 1.309n$.

The algorithm described in section 8 was used and it can be shown that sliding conditions prevail for this case, therefore Eq. 38 must be used to obtain the normal component of the impulse. However, acording to the proposed contact law, the direction t of the tangential impulse must be such that the energy dissipation during the impact is maximized. Eq. 61 clearly shows the dependency of T_L on t. A plot of the variation of T_L for the present 3D pendulum example, when t is considered to rotate an angle α anticlockwise from the direction of $v_t^{(-)}$ is shown in figure 9. The direction of t corresponding to a maximum energy dissipation is denoted by the angle α_m.It should be observed that for this direction a value of $\pi_n = 4.02$ N.sec is predicted which is 78% of the maximum normal imput (for $\alpha = 196.25$). α_0 and α_1 are two other important angle values. They define a sector corresponding to directions of the tangential impulse that imply energy gains, therefore are not accetable.

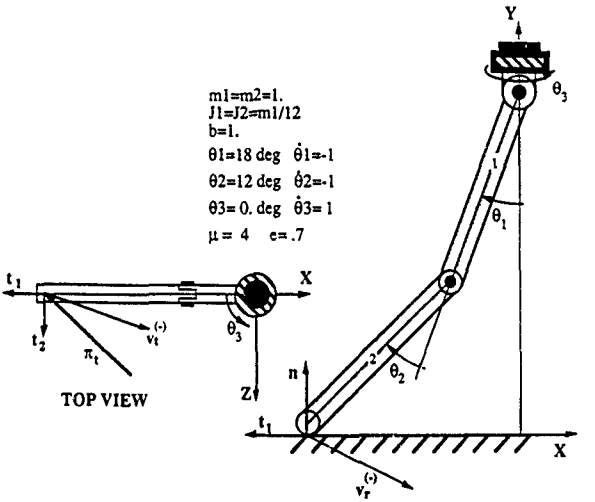


Figure 9. 3D Double pendulum

Figure 10 illustrates the vectors: $v_t^{(-)}$, the tangential velocity jump, Δv_t, the outgoing tangential velocity, $v_t^{(+)}$, and the tangential impulse, π_t, corresponding to maximum energy dissipation. Another vector t for the direction of the tangential impulse is also represented which was obtained from Ref. [16] where an averaging process of the tangential components of the approach and separation velocities have been used. It clearly shows that the tangential impulse, in general is not collinear with $v_t^{(-)}$ and $v_t^{(+)}$. The present calculations predict at the point of contact:

$$v_r^{(+)} = .675\, t_1 + .41\, t_2 + .916n$$
$$\Delta p = 1.26\, t_1 - 99\, t_2 + 4.026n$$

A modification on the algorithm in section 8 must then be introduced in step 3 where π_n must be found from Eq. 52 (38) for closed (open) loop systems, using a direction t that minimizes T_L.This process is highly non linear and can be carried out with the use of any standard minimization algorithm.

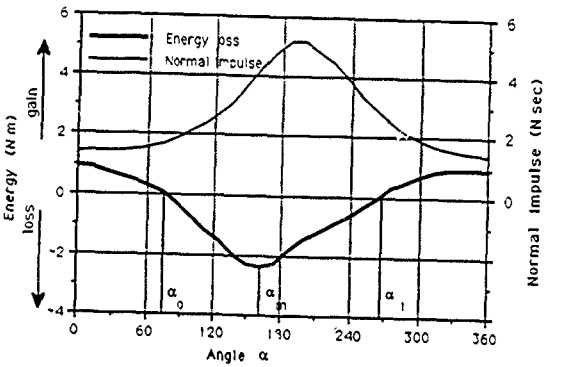


Figure 9. Energy loss and normal impulse versus t the direction of the tangential impulse

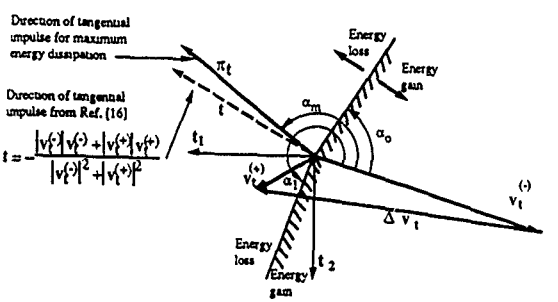


Figure 10. Impulse and rebound predictions for the 3D pendulum

10. Conclusions

In this paper, a method was presented for the piecewise analysis of the intermittent motion of mechanical systems. A canonical form of the impulse of momentum equations for multibody systems can be established in terms of a minimum set of joint coordinates and can be solved easily for the jump in the joint velocities during the impact. This solution also finds the impulsive forces exerted on the contact surfaces of the colliding bodies and provide a means by which necessary conditions are introduced for stiction or sliding to occur. The analytical set is completed by a system of contact laws involving the introduction of a coefficient of restitution traditionally used for the description of two body collisions and a coefficient of friction associated with the Coulomb's dry friction model and assuming that friction is developed in a such a way that a maximum dissipation of energy will be observed. Since the impulse-momentum represent a first integral of the motion, velocity jumps can be calculated in a straight forward manner allowing an immediate assessment of energy changes.

Acknowledgement

This research was supported by AGARD, project P77.

REFERENCES

1. Meirovich, L., *Methods of Analytical Dynamics*, McGraw-Hill Book Co., 1970.
2. Wittaker, E.T., *Analytical Dynamics*, 4th ed., Cambridge University Press, London, 1937.
3. Pars, L.A., *A Treatise of Analytical Dynamics*, William Heinemann, Ltd., London, 1965
4. Kane, T.R., *Dynamics*, Holt, Rinelaut and Wilson, Inc., 1968.
5. Brach, R.M., "Rigid Body Collisions," *ASME J. of Applied Mechanics*, Vol 56, pp. 133-138, 1989.
6. Haug, E.J., Wu, S.C., Yang, S.M., "Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact and Constraint Addition and Deletion - I," Mechanism and Machine Theory, Vol. 21, No. 5, pp 401-406, 1986.
7. Lankarani, M.H., Nikravesh, P.E., "Application of the Canonical Equations of Motion in Problems of Constrained Multibody Systems With Intermittent Motion," 14th ASME Design Automation Conf., *Advances in Design Automation*, DE-Vol. 14, pp. 417-423, 1988.
8. Nikravesh, P.E., *Computer Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1988
9. Wittenburg, J., *Dynamics of Systems of Rigid Bodies*, B. G. Teubner, Stuttgart, W Germany, 1977.

10. Jerkovsky, W., "The Structure of Multibody Dynamics Equations," *J. Guidance and Control* 1, pp173-183, 1978.

11 Kim, SD.S, Vanderploeg, M.J., "A General and Efficient Method for the Dynamic Analysis of Mechanical Systems using Velocity Transformations," *ASME J. Mech., Trans., Auto. in Design* 108, pp176-182, 1986.

12 Nikravesh, P.E., Gim, G., "Systematic Construction of the Equations of Motion for Multibody Systems containing Closed Kinematic Loops", *Proc., 15th Design Automation Conference*, Montreal, Canada, pp.27-33, ASME, New York, 1989.

13 Lanczos, C., *The Variational Principle of Mechanics*, Dover Publications, New York, 4th ed., 1986.

14 Jean, M., Moreau, J.J., "Unilaterality and Dry Friction in the Dynamics of Rigid Body Collections," *Proc. Contact Mechanics Int. Symp.*, Edt. A. Curnier, PPUR, pp31-48, 1992.

15. Jean, M., Moreau, J.J., "Dynamics of Elastics or Rigid Bodies With Frictional Contact: Numerical Methods," Mecanique, Modelisation Numerique et Dynamique des Materiaux, *Rencontres Scientifiques du Cinquantenaire*, 8-10 Avril 1991, pp9-29, 1991.

16. Smith, C.E., "Predicting Rebounds Using Rigid Body Dynamics," *Transactions of the ASME*, Vol. 58, pp754-758, 1991.

IMPACT DYNAMICS OF MULTIBODY MECHANICAL SYSTEMS AND APPLICATION TO CRASH RESPONSES OF AIRCRAFT OCCUPANT/ STRUCTURE

D. MA, R. MENON and H. M. LANKARANI
Mechanical Engineering Department
and National Institute for Aviation Research
Wichita State University
Wichita, KS 67260-0035, USA

ABSTRACT. Analytical evaluation of the performance of multibody mechanical systems becomes rapidly unmanageable as the complexity of the systems increase. For problems that involve intermittent motion due to an impact, prediction of the responses is even more difficult. In an impact, nonlinear contact forces of unknown nature are created, which act and disappear over a short period of time. In this paper, different contact force models are formulated, with which a continuous analysis method is developed for a simple two-particle impact. The procedure is then generalized to impact in multibody systems using the concept of effective mass. A piecewise analysis method is discussed, which is based on a canonical form of the system impulse/momentum equations. The suitability of these methods are discussed by application of these procedures to some examples. An optimization methodology is then discussed for the selection of proper parameters in a given contact force model. The use of this technique in the selection of the most suitable materials, which are impact-resistant, is also discussed.

The methods discussed earlier are applied to the seat-occupant-restraint system of an aircraft. A description of a computer-aided analysis environment, including a multibody model of the occupant and a nonlinear finite element model of the seat, is provided, which can be used to re-construct variety of crash scenarios. These detailed models are useful in studies of the potential human injuries in a crash environment, injuries to the head, the upper spinal column, and the lumbar area, and also structural behavior of the seat. The problem of reducing head injuries to an occupant in case of a head contact with the surroundings (bulkhead, interior walls, or instrument panels), is then considered. The head impact scenario is re-constructed using a nonlinear visco-elastic type contact force model. A measure of the optimal values for the bulkhead compliance and displacement requirements is obtained in order to keep the possibility of a head injury as little as possible. This information could in turn be used in the selection of suitable materials for the bulkhead, instrument panels, or interior walls of an aircraft. The developed analysis tool also allows aircraft designers/engineers to simulate a variety of crash events in order to obtain information on mechanisms of crash protection, designs of seats and safety features, and biodynamic responses of the occupants as related to possible injuries.

1. Introduction

A collision between the two bodies is known as an impact during which forces are created that act and disappear over a short period of time. The duration of the contact

period governs the choice of the method used to analyze the impact. The methods for predicting the impact responses can primarily be classified into two groups. In one, the impact is treated as a discontinuous event. Momentum-balance/impulse equations are usually formulated by integrating the acceleration-based form or the canonical form of the governing equations of motion. The solution to these equations gives the jump in the velocities or momenta, with which the integration of the equations of motion can be re-started. This classical approach has been followed by many, for direct-central and oblique impacts of two free solids not connected to other bodies [1-3]. For an impact within a multibody system, a set of momentum balance-impulse equations was developed by Wehage [4]. Khulief and Shabana extended these equations to flexible multibody systems [5]. In [6], it was shown that in order to gain numerical efficiency and stability, the equations of motion for a multibody system may be assembled in a canonical form. In this formulation, a reduced set of momentum balance-impulse equations was developed based on the time integration of a canonical form of the equations of motion. In order to perform this so-called "piecewise analysis," the coefficient of restitution between the two colliding bodies must be known. Although this method is relatively efficient, the unknown duration of contact limits its applicability. If the duration of contact is large enough that significant changes occur in the configuration of the multibody system, then the assumption of instantaneity of impact is no longer valid.

In the second group of methods used for impact analysis of multibody systems, the local deformations and the contact forces are treated as continuous. The impact analysis of the system is hence performed simply by including the contact forces in the system equations of motion during the contact period. The application of this so-called "continuous analysis," requires knowledge of the variation of contact force during the contact period. Different models have been postulated to represent this variation. In the most simple one, the contact force is modeled by a parallel linear spring-damper element [7]. This model, known as the Kelvin-Voigt viscoelastic model, has been used for the impact between two bodies within a multibody system [8]. However, this linear model may not be very accurate since it does not represent the overall nonlinear nature of an impact. Furthermore, the half-sine shape solution that it provides for the local deformation of the two bodies in the direction of impact suggests that the two bodies exert tension on each other right before the rebounding stage. A more suitable model of the contact force is the nonlinear Hertzian force-displacement law [9]. Although Hertzian theory is based on elasticity, some studies have been performed to extend the theory to include energy dissipation [7, 10-12].

In this paper, continuous contact force models are presented, for which unknown parameters are analytically evaluated. In one model, internal damping of the impacting bodies is used to represent the energy dissipation at low impact velocities. A hysteresis damping function of this nature assumes that the loss of energy in impact is all due to the material damping of the colliding solids, which dissipates energy in the form of heat. At fairly moderate or higher velocities of colliding solids, especially metallic solids, permanent indentations are left behind on the colliding surfaces. In the second contact force model, which covers these cases, local plasticity of the surfaces in contact becomes the dominant source of energy dissipation in impact. For both models, the unknown parameters are evaluated by energy and momentum considerations in terms of the velocities of the solids before impact and the coefficient of restitution. The two-particle model is then generalized to impact analysis between two bodies of a multibody mechanical system. The concept of effective masses is introduced to take into account the effect of joint forces or impulses. The suitability of the different methods is discussed by applications of the procedures to some examples, including a vehicle rollover. An optimization methodology is then discussed for the selection of proper

parameters in a given contact force model. The objective would be to minimize the maximum acceleration of the system effective mass(es), such that the maximum deformations fall below a certain range. The design parameters would be, for example, the linear or nonlinear stiffness coefficients, damping coefficients, amount of permanent indentation, etc. The use of this technique in the selection of the most suitable materials, which are impact-resistant, are discussed.

The methods discussed earlier are applied to the seat-occupant-restraint system of an aircraft. To improve passenger safety, the design and the arrangement of the seats, types and characteristics of the restraints, and the dynamic performance of the occupants in various crash environments must be known. A description of a computer-aided analysis environment is provided, which can be used to re-construct variety of crash scenarios. The analysis capability relies on the development of a multibody dynamic model for the occupant. The occupant model is based on the model from the code SOM-LA/TA (Seat Occupant Model-Light Aircraft/Transport Aircraft) [13-14]. Body segments are connected by joints which impose resistance in terms of a nonlinear displacement limiting moment, viscous damping, and muscle tone. The model of the upper spinal column has been refined to include the seven vertebrae and a hydrodynamic model of the skull/brain. A quasi-static analysis capability of the rigid multibody dynamics of the gross motion of the occupant, together with a finite element model of the lumbar spine, have also been developed. These detailed descriptions make the model useful in studies of the potential human injuries in a crash environment, injuries to the head, the upper spinal column, and the lumbar area. A nonlinear finite-element model of the seat structure is also used which includes large plastic deformations, buckling of bending members, material nonlinearity, etc.

One aspect of these studies is to reduce head injuries to an occupant in case of a head contact with the surroundings (bulkhead, interior walls, or instrument panels). In this paper, the head impact scenario was re-constructed using a nonlinear visco-elastic type contact force model of exponential form. The stiffness coefficients in the model were obtained from experimental correlations, and the damping coefficient was obtained using Rayleigh formulation. Correlated studies of the analytical simulation with impact sled test results, performed at National Institute for Aviation Research (NIAR) and Civil AeroMedical Institute (CAMI), were accomplished. A parametric study of the coefficients in the contact force model was then performed in order to obtain the correlation between the coefficient and the Head Injury Criteria. A measure of the optimal values for the bulkhead compliance and displacement requirements was thus obtained in order to keep the possibility of a head injury as little as possible.

2. Contact Force Models

When two solids are in contact, deformation takes place in the local contact zone resulting in a contact force. This suggests that the contact force is directly related to the amount of local deformation or indentation of the two solids. The best-known force model for the contact between two spheres of isotropic material was developed by Hertz based on the theory of elasticity [9]. With radii R_i and R_j of the two spheres "i" and "j", and masses m_i and m_j, the contact force f follows the relation

$$f = K \delta^n , \qquad (1)$$

where δ is amount of the relative penetration or indentation between the surfaces of the two spheres and $n = 1.5$. The generalized parameter K depends on the material properties and the radii of the spheres:

$$K = \frac{4}{3\pi(h_i + h_j)} \sqrt{\frac{R_i R_j}{R_i + R_j}} , \qquad (2)$$

where the material parameters h_i and h_j are

$$h_l = \frac{1 - \nu_l^2}{\pi E_l} , \quad l = i, j \qquad (3)$$

Variables ν_l and E_l are, respectively, the Poisson's ratio and the Young's modulus associated with each sphere.

Consider now a situation for which the contact between the two spheres is caused by a collision. The two spheres have velocities $V_i^{(-)}$ and $V_j^{(-)}$ right before impact. It is intended to determine the variations of the interaction force between the two spheres during the short period of contact. The normal direction n to the contact surfaces and a pair of forces f and -f are shown in Figure 1. In general, the resulting contact may be considered to occur in two phases: the compression phase and the restitution phase. During the compression phase, the two spheres deform in the normal direction to the impact surface, and the relative velocity of the two spheres is reduced to zero. The end of the compression phase is referred to as the instant of maximum compression, or maximum approach. The restitution phase starts at this point and lasts until the two spheres separate.

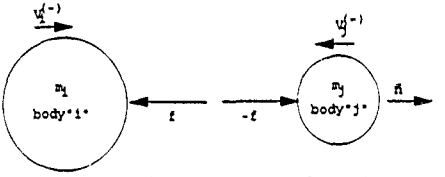


Figure 1. A direct-central impact of two spheres.

Generally, the two spheres will not rebound with the same initial velocities, because part of the initial kinetic energy is dissipated in the form of permanent deformation, heat, etc. It is apparent that the contact force model of equation (1) cannot be used during both phases of contact, since this would suggest that no energy is dissipated in the process of impact. One popular treatment is based on the idea that dissipation of energy occurs in the form of internal damping of colliding solids. This assumption is valid for low impact velocities; i.e., those impact situations for which impact velocities are negligible compared to the propagation speed of deformation waves across the solids. The contact force model will then be in terms of a damping coefficient D,

$$f = K \delta^n + D \dot{\delta} , \qquad (4)$$

where $\dot{\delta}$ is the relative (or indentation) velocity of the solids. A hysteresis form for the damping coefficient was proposed by Hunt and Grossley [10] as

$$D = \mu \delta^n , \qquad (5)$$

where the parameters μ is called the "hysteresis damping factor." The contact force model of equation (4) may be used for the entire period of contact. With this model, the energy loss is assumed to be due to the material damping of the bodies, which would

dissipate energy in the form of heat. For known parameters K and D (or μ), the shape of the hysteresis loop corresponding to this force model and the solution corresponding to the variations of the indentation with time are shown in Figure 2 [15]. In this figure, $t^{(\cdot)}$, $t^{(m)}$, and $t^{(+)}$ denote the time of initial contact, the time of maximum indentation, and the time of separation of the local contact surfaces, respectively. Variables δ_m and f_m refer to the values of indentation and the contact force at time $t^{(m)}$. In the contact force model of equation (4), the damping coefficient D or the hysteresis damping factor μ must be determined. An estimate of the parameter μ based on the classical impulse-momentum equation and the work-energy principle can be determined.

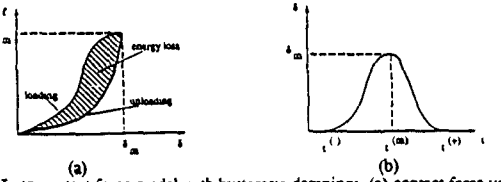


Figure 2. Hertz contact force model with hysteresis damping: (a) contact force versus indentation and (b) indentation versus time.

From consideration of the kinetic energies before and after impact, the energy loss ΔT may be expressed in terms of the coefficient of restitution e and the relative approach velocity $\dot{\delta}^{(\cdot)} = V_i^{(\cdot)} - V_j^{(\cdot)}$ as

$$\Delta T = \frac{1}{2} m^{(eq)} \dot{\delta}^{(\cdot)^2} (1 - e^2) , \tag{6}$$

where

$$m^{(eq)} = \frac{m_i m_j}{m_i + m_j} , \tag{7}$$

is the system equivalent mass. The energy loss may also be expressed by integration of the contact force around the hysteresis loop as [16]

$$\Delta T = \oint D\dot{\delta}\, d\delta = \oint \mu\, \delta^n \dot{\delta}\, d\delta \cong 2 \int_0^{\delta_m} \mu\, \delta^n \dot{\delta}\, d\delta = \frac{2}{3} \frac{\mu}{K} m^{(eq)} \dot{\delta}^{(\cdot)^3} , \tag{8}$$

where $\oint$ refers to the integration around a hysteresis loop for a contact force of the form shown in Figure 2(a). The hysteresis damping factor μ may be evaluated by comparing the right sides of equations (6) and (8),

$$\mu = \frac{3K(1 - e^2)}{4\dot{\delta}^{(\cdot)}} , \tag{9}$$

which shows a direct relationship between the coefficient of restitution and an equivalent damping factor. The contact force in conjunction with the damping representation may be written in an alternative form as

$$f = K \delta^n \left[1 + \frac{3(1 - e^2)}{4} \frac{\dot{\delta}}{\dot{\delta}^{(\cdot)}} \right] , \tag{10}$$

which shows a direct relationship between the coefficient of restitution and the contact force.

The clear advantage of the Hertz contact force model, $f = K\delta^n$, with its damping representation in equation (10) over the Kelvin-Voigt viscoelastic model is its nonlinearity. The overall pattern of impact is far from linear, while the Kelvin-Voigt model and its damping representation are linear models. The solution for indentation corresponding to the linear models is a half-damped harmonic. This indicates that the bodies in impact must exert tension on each other right before separation. On the other hand, the Hertzian contact force model predicts no tension on the bodies before separation, as observed from the solution for its corresponding indentation of Figure 2(b).

At higher impact velocities, the dissipation of energy is mostly in the form of local plasticity. This means that some permanent indentation is left behind on the surfaces of the two spheres after separation, and this indication accounts for the energy loss in impact. This is not an unreasonable assumption for impact problems in which two metallic bodies with initial relative velocity larger than $10^{-5}\sqrt{E/\rho}$ collide [17], where ρ is the mass density and the quantity $\sqrt{E/\rho}$ is the larger of two propagation speeds of the elastic deformation waves in the colliding solids. With this condition, the contact force loads according to equation (1) during the compression period, and in loads according to

$$f = f_m \left[\frac{\delta - \delta_p}{\delta_m - \delta_p}\right]^n \quad \text{during restitution,} \tag{11}$$

where variable δ_p is the permanent indentation of the two spheres after separation. The shape of the hysteresis loop corresponding to this contact force model and the solution corresponding to the variation of the indentation with time are shown in Figure 3.

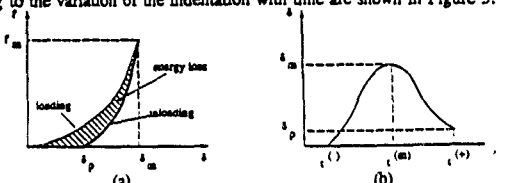


Figure 3. Hertz contact force model with permanent indentation: (a) contact force versus time and (b) indentation versus time.

The proposed contact force model can be used for the impact between two spheres, if the parameters in the model are known. The generalized parameter K may be evaluated from the radii and the material properties of the two spheres using equation (2). The remaining parameters are δ_m, f_m, δ_p, which can be determined by integrating the relative indentation equations of motion twice, substitution in the contact force expression, and integrating the contact force around the hysteresis loop and equating it to the kinetic energy loss, as [15]

$$\delta_m = \left[\frac{n+1}{2K}\, m^{(eq)}\, \delta^{(-)}\right]^{\frac{1}{(n+1)}} \tag{12}$$

$$f_m = K\,\delta_m^n \tag{13}$$

$$\delta_p = \frac{(n+1)\, m^{(eq)}\, \dot{\delta}^{(-)}}{2\, f_m} (1 - e^2) \qquad (14)$$

Hence maximum indentation of the two spheres and maximum contact force depend on the material properties, masses, radii, and velocities of the two spheres right before impact. The permanent indentation is evaluated from the initial approach velocities of the spheres and a known coefficient of restitution between the spheres.

A continuous analysis may now be performed by numerically integrating the equations of motion of the two spheres forward in time in conjunction with the developed contact force model. A solution is thus obtained in the form of positions, velocities, and accelerations of the spheres at any instant of time during the contact period. As a by-product of the preceding parameter identification process, one can approximate the duration of the contact period between the two spheres as [18]

$$\Delta t \cong 2.94\, \frac{\delta_m}{\dot{\delta}^{(-)}} . \qquad (15)$$

To illustrate the parameter estimation process, a numerical example is considered here. For a direct-central impact of two identical aluminum spheres, a coefficient of restitution of 0.7 and a duration of contact of 135 μs were recorded from experiment [7]. Both spheres had equal and opposite impact velocities of 0.15 m/s, and each had a radius of 0.02 m.

Based on the presented theory, it is intended to construct the contact force model between the two spheres. The initial indentation velocity between the two spheres is $\dot{\delta}^{(-)} = 0.3$ m/s. The speed of deformation waves is 2.6 (10^3) m/s, which provides a limiting value of 0.026 m/s for the impact to be considered elastic. Hence, the Hertz contact force model with permanent indentation is a valid one. The generalized parameter K is calculated from equation (2), with $\nu = 0.33$, to be equal to 5.50(10^9) N/m$^{1.5}$. The equivalent mass of the two spheres is obtained from equation (7) as $m^{(eq)}$ = 0.046 kg. From equations (12), (13), and (14), the unknown parameters in the contact force model are evaluated as

$$\delta_m = 1.55(10^{-5})\ \text{m}; \quad f_m = 336\ \text{N}; \quad \delta_p = 7.92(10^{-6})\ \text{m}.$$

With all the parameters determined, the contact force model of the two spheres is constructed as shown in Figure 4.

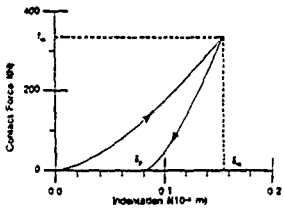
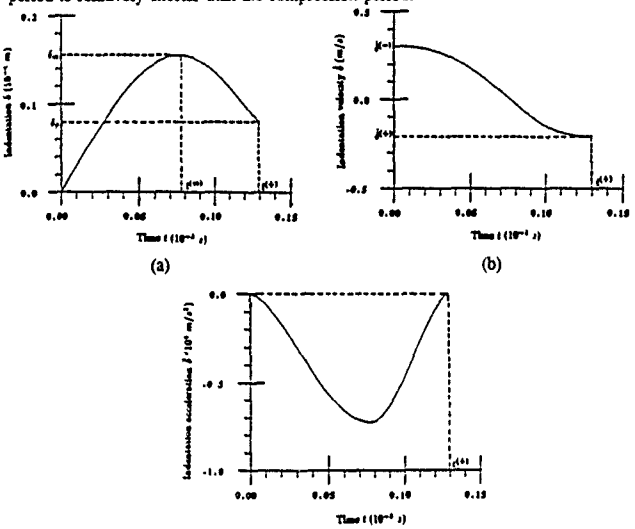


Figure 4. A Hertzian contact force model with permanent indentation for the impact of two identical aluminum spheres with impact velocities of 0.15 m/s each.

Note that the permanent indentation is slightly larger than half of the maximum relative indentation for this impact situation, which is a significant amount. This

suggests that the local plasticity of the contact surfaces during impact cannot be neglected, and it again justifies the validity of the Hertzian contact force model with permanent indentation. A continuous analysis may be performed on the two spheres with the constructed contact force model. The relative indentation equation of motion, $m^{(eq)}\ddot{\delta} = -\hat{f}$, with its corresponding initial conditions, and the developed contact force model of Figure 4 are integrated forward in time until the separation time. This results in a relative departing velocity, $\dot{\delta}^{(+)}$, of 0.21 m/s and a duration of contact of 130 μs, which is very close to the experimental value of 135 μs.

The solution for variations of indentation, indentation velocity, and indentation acceleration versus time is shown in Figure 5. If a piecewise analysis with the coefficient of restitution of 0.7 is performed, the relative departing velocity of the two spheres is obtained as $\dot{\delta}^{(+)} = -e\dot{\delta}^{(-)} = -0.21$ m/s. Comparison with $\dot{\delta}^{(+)}$ from Figure 5(b) shows that the results from a continuous analysis and a piecewise analysis match closely. The energy dissipated in impact can also be evaluated as $\Delta T = 1.06(10^{-3})$ J. Also, the approximate expression of equation (15) for the duration of contact results in a value of $\Delta t = t^{(+)} - t^{(-)} \cong 152$ μs. The apparent difference between this approximate value and of the experimental value of 135 μs, is due to the fact that the restitution period is relatively shorter than the compression period.



(a) (b)

(c)

Figure 5. Results from a continuous analysis of the two aluminum-spheres impact. Time variations of: (a) relative indentation, (b) relative indentation velocity, and (c) relative indentation acceleration.

3. Generalization to Multibody Impact

We now generalize the preceding discussion to an impact situation within a multibody system. Let the two colliding bodies be "i" and "j" in the system of Figure 6. The points of contact on the two bodies are P_i and P_j, and $\mathbf{n}$ is a unit vector in the normal direction to the contact surfaces of the two bodies. No matter which type of coordinates are used to assemble the equations of motion for the multibody system, the coordinates and the velocities of the bodies can be calculated, at any instant of time, from the solution of the equations of motion. For a known system configuration at the initial time of contact, the location of the contact points $\mathbf{r}_i^{P(-)}$ and $\mathbf{r}_j^{P(-)}$ and the components of the algebraic unit vector $\mathbf{n}$, with respect to a non-moving xyz coordinate system, may be calculated. From the known rigid body velocities, velocities of the contact points in the xyz coordinate system, $\dot{\mathbf{r}}_i^{P(-)}$ and $\dot{\mathbf{r}}_j^{P(-)}$, may also be calculated at that time. Hence, the indentation and the indentation velocity at the initial time of contact are

$$\delta^{(-)} = 0 \tag{16}$$

$$\dot{\delta}^{(-)} = \mathbf{n}^T \left[\dot{\mathbf{r}}_i^{P(-)} - \dot{\mathbf{r}}_j^{P(-)} \right] , \tag{17}$$

in which the symbol T performs the transpose operation and velocities of the contact points are projected in the normal direction to the contact surface. The expression in equation (2) for the parameter K can be used for the contact between any two bodies if the local surfaces of contact are both spherically shaped. Similar expressions have been obtained by Hertz [9] and others [7, 19] for other shapes of the local contact surfaces such as sphere on plane, parallel cylinders, and plane on plane. Once the generalized parameter K is calculated, with a given coefficient of restitution e and the initial approach velocity $\dot{\delta}^{(-)}$, all other parameters in the contact force model can be determined.

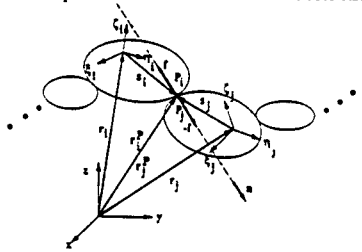


Figure 6. Impact within a multibody system

With known variations of the contact force during the contact period, a continuous analysis of the system can be performed simply by adding these forces to the multibody system equations of motion. This analysis method provides accurate results, since all of the equations of motion are integrated over the period of contact. It thus accounts for the changes in the configuration and the velocities of the system during that period. To avoid computational inaccuracies and inefficiencies associated with the integration of the system equations of motion over the period of contact, scaling of the time axis is sometimes performed.

In order to gain some computational efficiency, we may use an alternative method

which requires some approximation, that the duration of contact period is small enough such that the configuration of the system, and therefore, the normal direction of impact remains the same before and after impact. The purpose of this analysis is to represent the two bodies "i" and "j" in Figure 6 by two equivalent particles of Figure 1. To account for the inertia properties of the colliding bodies when the equivalent particle model is used, one must evaluate an effective mass $m_l^{(eff)}$ for each body, $l = i, j$. Since the body "l" may be kinematically constrained to their bodies in the system, the reaction forces at the kinematic joints affect the motion of the body during the contact period.

The compensation for the effects of joint forces in the evaluation of effective mass is based upon whether an energy or a Newtonian approach is followed. Using an energy approach, the effective mass of body "l" is evaluated as

$$m_l^{(eff)} = m_l \frac{T^{(\omega)}}{T_l} \quad ; \quad l = i, j \tag{18}$$

where m_l and T_l are the mass and kinetic energy of body "l", and $T^{(\omega)}$ denotes the total kinetic energy of all the bodies that are connected directly or indirectly to body "l" including body "l" [8]. This collection of bodies is referred to as a sub-system associated with body "l". Note that the kinetic energies in equation (18) are only due to the components of velocity of the bodies in the direction of impact. As a direct consequence of Newton's second law of motion, the effective mass of a body must be a measure of the resistance of the body toward its velocity change. Hence in a Newtonian approach, the effective mass of each colliding body "l" is evaluated in such a way that the inertial forces acting on the body, in the direction of impact, would be equivalent to the inertial forces acting on the sub-system containing body "l" in the direction of impact; i.e.,

$$m_l^{(eff)} = \frac{\sum_k \mathbf{n}^T m_k \ddot{\mathbf{r}}_k}{\mathbf{n}^T \ddot{\mathbf{r}}_l} \quad ; \quad l = i, j \tag{19}$$

where m_k and $\ddot{\mathbf{r}}_k$ are the mass and the translational acceleration of body k, and the summation k is over all of the bodies in the sub-system. Note that the effective masses obtained by the Newtonian approach may or may not be the same as the effective masses obtained from the energy approach.

Once the calculation of effective masses of the two colliding bodies is performed right before impact (at the time of initial contact), the direct-central impact model of the two spheres is reconstructed with effective masses $m_i^{(eff)}$ and $m_j^{(eff)}$ and initial velocities $\mathbf{n}^T \dot{\mathbf{r}}_i^{(-)}$ and $\mathbf{n}^T \dot{\mathbf{r}}_j^{(-)}$. The relative indentation equation of motion of the two spheres during the period of contact then becomes

$$m^{(\omega)} \ddot{\delta} = -f \tag{20}$$

where $\ddot{\delta} = \mathbf{n}^T (\ddot{\mathbf{r}}_i^P - \ddot{\mathbf{r}}_j^P)$, and

$$m^{(\omega)} = \frac{m_i^{(eff)} m_j^{(eff)}}{m_i^{(eff)} + m_j^{(eff)}} \tag{21}$$

Equation (20) is a second-order ordinary differential equation that may be solved numerically during the period of contact knowing the variations of the contact force. A proper set of initial conditions at the time of initial contact $t^{(}$ is provided by equations (16) and (17). The numerical integration of equation (20) with the developed contact force model is carried out until the surfaces of the two spheres modeling the two bodies "i" and "j" separate from each other.

For illustration of impact analysis of a multibody system, a study of a vehicle

rollover was performed. Simulations were performed to duplicate experimental rollover test of a vehicle at 30 mph on a FMVSS 208 rollover cart impacting a water-filled decelerator system, thereby throwing the vehicle off the cart. The initial roll angle was 23 degrees. The vehicle, at the time of departure, had a velocity of approximately 25 mph in the translational direction and 1.5 rad/s in the roll direction as shown in Figure 7. A number of different analyses with rigid and flexible bodies have been performed on this vehicle [20, 21]. Two additional simulations related to the topic of this paper were also performed and the results were compared to those from the experiment.

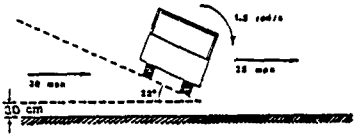


Figure 7. Initial configuration of the vehicle for the rollover test.

In the first simulation, a piecewise analysis was performed for any contact between the vehicle rollbar cage and the ground. A set of canonical impulse-momentum equations was used at the time of impact [6]. At the time of impact $t^{(-)}$, integration of the system equations of motion was interpreted and the jump in the system momenta were calculated. With new momenta, integration of the system equations of motion were restarted. This procedure can be summarized as follows. First the linear system,

$$\left(\mathbf{D}^{*}\mathbf{M}^{-1}\mathbf{D}^{*T} \right) \begin{bmatrix} \Delta\sigma \\ \boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} 0 \\ -(e+1)\mathbf{N}\dot{\mathbf{q}}^{(-)} \end{bmatrix} \tag{22}$$

was solved for $\Delta\sigma$ and $\boldsymbol{\pi}$, and then the momenta after impact was updated as:

$$\mathbf{p}^{(+)} = \mathbf{p}^{(-)} + \mathbf{N}^{T}\boldsymbol{\pi} \tag{23}$$

In these equations, (-) and (+) refer again to the quantities before and after impact respectively, $\mathbf{N}$ is a composite normal vector, e the coefficient of restitution, $\mathbf{q}$ the velocity vector, $\boldsymbol{\pi}$ the impulse due to the contact force, σ the Lagrange multipliers associated with velocity constraints, $\mathbf{M}$ the system mass matrix, $\mathbf{D}^{*} = [\,\mathbf{D}^{T}, \mathbf{N}^{T}\,]^{T}$, $\mathbf{D}$ the constraints Jacobian matrix, and $\mathbf{p}$ the vector of system momenta.

A continuous analysis was then performed using the Hertzian contact force model with permanent indentation. To develop the contact law, the curvatures of the rollbar at the area of contact was used, as shown in Figure 8 [19]. The parameters in the contact force model were determined, and the resulting model is shown in Figure 9. The solutions for relative indentation, velocity, and acceleration of the rollbars/ground may be determined similar to the procedure for the two aluminum spheres impact. The configuration of the vehicle during the rollover is shown in Figure 10. Summary of the results from the two analyses and the experiment is shown in Table 1. As observed, although piecewise analysis is computationally more efficient, the corresponding results are not very accurate. This is due to the fact that the duration of the contact is large enough (approximately ¼ second) that significant changes occur in the configuration of the system. In comparison however, the continuous analysis with the developed contact model resulted in much more closer results to the experimental values.

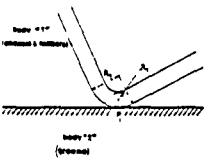
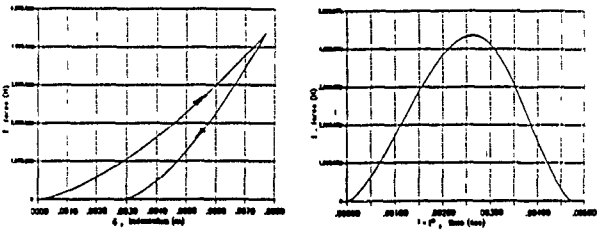


Figure 8. Curvature used to develop the contact force model for rollbar/ground contact.



(a) (b)

Figure 9. Shape of the contact force model for the rollbar/ground contact: (a) force versus indentation, (b) force versus time.

Figure 10. Rollover configuration of the vehicle.

Table 1. Summary of the results from experiment and analyses of the vehicle rollover.

Variables	Piecewise	Continuous	Experiment
Vertical acceleration (g)	5	17	15
First contact duration (s)	0	0.15	0 25
Settling time (s)	2.8	3.0	3.0
CPU ratio	1.0	2.7	—

4. Parameter Optimization of the Contact Force Model

Reducing the severity of an impact to a system is a significant aspect of the engineering design. This requires reduction of the contact forces and their effect. An optimization methodology can be formulated for the selection of the proper parameters in the contact

force model so as to minimize the maximum value of the contact force. Consider the two-particle model of any impact, shown in Figure 1, for which a single differential equation of motion in the direction of relative indentation of the contact surfaces, equation (20) has been written. The optimization methodology was formulated as:

$$\text{Objective Function} : \quad \text{Minimize} \mid \ddot{\delta}_{max} \mid \qquad (24)$$
$$\text{Subject To} : \quad \delta \leq \Delta$$

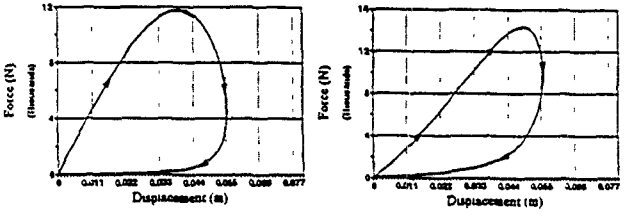
where $\ddot{\delta}_{max}$ represents the maximum acceleration of the system equivalent mass under the action of the contact force, and Δ is a limit on the amount of relative indentation. A solution to this problem was obtained using the method of modified feasible direction for constrained minimization. A numerical integrator was used at every design iteration to obtain the system dynamic response for a given set of design variables.

As an example, the preceding steps were applied to a vibro-impact system of equivalent mass of 6 kg, initial relative impact velocity of 0.5 m/s, and a limit of 5 5 cm on the amount of relative indentation. The nonlinear Hertzian viscoelastic model of equation (4) was used to represent the contact force. The optimization problem of equation (24) was solved for the system, for which the design variables were the parameters K and μ in the contact force model. The optimized parameters were determined as:

$$K = 685.7 \quad N/m^{1.5}; \quad \mu = 9{,}428.6 \quad N.s/m^{2.5}$$

under which the resulting peak value of maximum acceleration, $\mid \ddot{\delta}_{max} \mid$ was 4.2 m/s^2 Diagrams for the contact force versus indentation, when optimal K and μ are used, are shown in Figure 11(a). A sample diagram for non-optimal parameters is also shown in Figure 11(b). It can be observed that the peak contact force is greatly reduced for the optimal case. Also, the system with optimal parameters tends to dissipate more energy than the non-optimal one during the period of the contact.

To study the effect of initial impact velocity, the optimization problem was then solved with a few different initial velocities. Table 2 summarizes some of the results, from which it can be observed that at higher velocities both stiffness and damping characteristics must increase in order to reduce the maximum acceleration and to keep the amount of indentation below a certain limit. The resulting peak acceleration also increases for increasing initial impact velocity. As observed, the optimization procedure can provide useful information on the compliance requirement of crashworthy materials.



(a) (b)

Figure 11 Force versus displacement diagram for the system with (a) optimal parameters, (b) non-optimal parameters.

Table 2. Optimal Coefficients for the nonlinear Hertzian Viscoelastic model

Case No	V_o (m/s)	K ($N/m^{1.5}$)	μ ($N.s/m^{2.5}$)	$\lvert \delta_{max} \rvert$ (m/s^2)
1	0.5	685 7	9428 6	4.22
2	1 0	2285.7	20000	16 89
3	1.5	9142.9	24285.7	38.41

5. Applications to Crash Responses of Aircraft Occupant/Structure

As an application of the theory discussed earlier, the crash responses of aircraft occupant/structure will be presented. To improve aircraft crash safety, conditions critical to occupants survival during a crash must be known. In view of the importance of this problem, studies of post-crash dynamic behavior of victims are necessary in order to reduce severe injuries. In this study, crash dynamics program SOM-LA/TA (Seat Occupant Model - Light Aircraft / Transport Aircraft) was used [13,14]. Modifications were performed in the program for reconstruction of an occupant's head impact with the interior walls or bulkhead. A viscoelastic-type contact force model of exponential form was used to represent the compliance characteristics of the bulkhead. Correlated studies of analytical simulations with impact sled test results were accomplished. A parametric study of the coefficients in the contact force model was then performed in order to obtain the correlations between the coefficients and the Head Injury Criteria. A measure of optimal values for the bulkhead compliance and displacement requirements was thus achieved in order to keep the possibility of a head injury as little as possible. This information could in turn be used in the selection of suitable materials for the bulkhead, instrument panel, or interior walls of an aircraft. Before introducing the contact force model representing the occupant head impacting the interior walls, descriptions of impact sled test facilities, multibody dynamics and finite element models of the occupant/seat/restraint system, duplication of experiments, and measure of head injury are provided.

5.1 IMPACT SLED TEST FACILITIES

In order to evaluate the dynamic performances of the occupants and seats, the impact sled facilities at National Institute for Aviation Research (NIAR) are used. The horizontal impact test sled moves on a 23.8 meter track and has a maximum speed of 24.5 m/s. It runs on plastic shoes and is pneumatically propelled. The sled can attain its maximum acceleration within fifty feet of its travel. Steel straps are placed at the end of the track to bring the motion of the sled to an end as shown in Figure 12.

Figure 12. Impact Test Sled at NIAR.

Anthropomorphic dummies are used as occupants in the sled test experiments. There are standard 50th percentile (part 572) dummies, which represent an average male of weight 79.4 kg and height 1.75 meters. A 95th percentile dummy is also available, which represents an average male who weighs 90.7 kg and has a height of 1.88 meters. These dummies have openings in their head, vertebral column (lumbar) and femur. By fixing accelerometers and transducers inside these openings, the acceleration responses of head, lumbar, and femur can be determined during an impact test.

To conduct an impact sled test, the dummy is first restrained to a seat with a suitable restraint system and the seat is mounted on the sled. The sled is allowed to accelerate to the required value and then it coasts. The impact then occurs in the form of a deceleration pulse produced by the sled probe colliding with a number of steel straps. In addition to the data from the accelerometers, the displacement, velocity and acceleration of the dummy are recorded and evaluated using a high speed video system and optical target tracking techniques. Visible marks (targets) are attached on different parts of the dummy, seat and sled. During the motion of the sled, the targets are tracked by a video camera. An Ektapro 1000 motion analyzer is used to record the motion of the targets. The analyzer is capable of recording 1000 frames per second. From the resulting photographs, the path of motion of the targets can be determined. A software package called Motion Pro is utilized to digitize the motion recorded from the high speed video cameras by collecting coordinates of the points, centroids, angles and line segments. To overcome the difficulties of manual target tracking and also reduction of noise in the collected date, a simple automated target tracking procedure has also been developed [22].

5.2 MULTIBODY DYNAMICS MODEL OF THE OCCUPANT

Program SOM-LA/TA incorporates a dynamic model of human body and a finite element model of the seat structure. The program is intended to provide the designer with a tool which can analyze the structural elements of the seat as well as the dynamic response of an occupant during a crash. Both two- and three-dimensional occupant models are available. The user can select the type of occupant from the available options, which are: standard 50th percentile male human, standard 50th percentile dummy (part 572), nonstandard human, and nonstandard dummy. The two-dimensional model, which was investigated in this study, is represented by 11 rigid bodies/segments, and constitutes 11 degrees-of-freedom (Figure 13). The segments represent the head, upper and lower torso, pelvis, upper arms, forearms, thighs, and lower legs. The model is also configured to include beam elements in both the torso and the neck. This provides a measure of the vertical loads experienced by the occupant, in aircraft accidents.

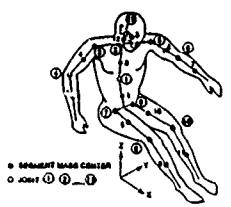


Figure 13. Two-dimensional model of the occupant [13].

The rotor cup, elbow, pelvic, knee, and head/neck joints are hinge-type connections. Rotations at the joints are resisted by torsional spring-dampers whose characteristics depend on the user selection of human or dummy occupant. External loads are applied to the body segments by the seat cushion, the floor, and by the belt restraints. Each of the forces acts normal to the contact surface. Friction forces are also applied by the seat bottom cushion and the floor, and in a direction opposite to the tangential component of relative velocity between the occupant segment and cushion or floor surface. Different types of restraint systems can also be modeled. They are lap belt, only attached to either the airframe or the seat, diagonal shoulder belt over the right or left shoulder attached to either the airframe or the seat, double shoulder belt, and double shoulder belt and lap belt tiedown strap. These exerted forces on the occupant are evaluated based on the geometrical (ellipsoidal, spherical, and cylindrical) shapes of the body segments. From these, the impact between the occupant and the aircraft interior can be predicted. A quasi-static equilibrium for finite element method in the incremental form is used to determine the seat structure responses. The seat is modeled with triangular plate elements, beam elements and springs. Large plastic deformations, as well as localized buckling of bending members can be predicted. Material nonlinearity is treated using elasto-plastic behavior and internal releases. At any time during the simulation, incremental displacements of the seat structure is obtained from the external forces and on updated tangential stiffness matrix. Total displacements of the seat are obtained by summation of all the incremental displacements. For dynamic formulation, a set of Lagrange's equations of motion in terms of independent or generalized coordinates is formulated and numerically generated. The differential equations of motion are converted to a set of first-order ordinary differential equations. The fourth order Runge-Kutta, as a starter and the predictor-corrector scheme of Adams-Bashforth/Adams-Moulton with adjustable step size is used as the numerical integrator.

In order to investigate the compliance and displacement characteristics of the aircraft bulkheads, an envelope was included for both the occupant and the seat. The parameters of the contact force model, as will be discussed later in this paper, were adjusted to obtain optimum compliance and displacement requirements.

Since the forces produced by the neck of the occupant model are also not clearly represented, further modifications are being carried out in order to replace the beam element of the neck with an efficient discretized neck model. Both two dimensional and three

dimensional models will be present. The two dimensional model has 9 degrees of freedom, and the three dimensional model has 36 degrees of freedom. The model includes the skull and the seven vertebral elements (cervical). The model shown in Figure 14, is based on the detailed cervicle model of Merrill et al. [23]. Another modification is construction of a brain model, based on the principles of hydrodynamics (squeeze effects). The basic model, representing the skull-brain interaction of Figure 14, studies the effects of two spherical bodies with fluid between them. For such a model, the pressure p and corresponding reaction force f developed inside are calculated using

$$p = \frac{3\mu_v V}{(c/R)^3 R\epsilon}\left[\frac{1}{1-\epsilon\cos(\theta)^4} - 1\right] \tag{25}$$

$$f = \frac{6\pi\mu_v VR}{(c/R)^3}\left[\frac{1}{\epsilon^3}\ln(1-\epsilon) + \frac{1}{1-\epsilon} - \frac{1}{2\epsilon}\right] \tag{26}$$

where μ, is the viscosity of the fluid; V the velocity of the inner spherical body; R the radius of the inner spherical body; ϵ the eccentricity ratio; c the radial clearance, and θ the angular coordinate measured from the maximum film thickness.

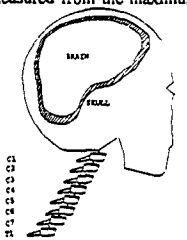


Figure 14. Modified skull-brain-neck model

An important measure of injury to an occupant in an aircraft accident is the amount of load transferred to the occupant lumbar spine [24]. The two dimensional occupant model utilizes a typical load-deflection formula of an intact lumbar spine vertebral column to predict the axial load and moments acting on the lumbar. This results in an incorrect representation of the dynamic responses produced by the spinal column during an impact. In order to predict the internal forces and moments in the spine, a finite element model has been incorporated into the occupant model. To overcome the problems of instability and inefficiency, a quasi-static methodology has been developed, which combines the finite element methods with rigid multibody dynamics principles. A description of this methodology is provided next.

In finite element analysis, the loads will generally be thought of as being applied, while the displacements are thought of as resulting. However, in analysis of multibody responses, it often happens that we have mixed boundary conditions. An example of a multibody system consisting of three bodies, two rigid bodies connected by a flexible body, is shown in Figure 15. Here, body "i" represents the pelvis, body "j" the thorax, and body "k" the lumbar spine.

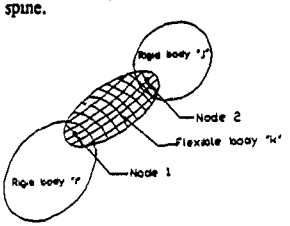


Figure 15. Flexible body connecting two rigid bodies.

While performing a dynamic analysis, the configuration of the rigid bodies and hence the displacements of nodes 1 and 2 of the flexible body are known. The problem is to find the deformed shape of the entire structure and also the forces and moments

acting on the rigid bodies by the deformable body. The quasi-static approach is formulated by rearranging external forces, displacements, and the structural stiffness matrix. To solve the structural equilibrium equations for the unknown forces and displacements, let us partition the known (k) and unknown (u) variables as

$$\begin{bmatrix} F_u \\ F_k \end{bmatrix} = \begin{bmatrix} K_{kk} & K_{ku} \\ K_{uk} & K_{uu} \end{bmatrix} \begin{bmatrix} \delta q_k \\ \delta q_u \end{bmatrix} \tag{27}$$

where matrices K_{kk}, K_{ku}, K_{uk} and K_{uu} are sub-matrices of rearranged structural stiffness matrix K, F_k is a vector of known forces acting on the spine, F_u a vector of unknown forces acting at the ends of lumbar spine, δq_u a vector of unknown displacements of lumbar spine, and δq_k known displacements of nodes 1 and 2 calculated from the rigid body dynamics. The unknown displacements and forces can be determined as

$$\delta q_u = K_{uu}^{-1}(F_k - K_{uk}\, \delta q_k) \tag{28}$$

$$F_u = K_{kk}\delta q_k + K_{ku}\delta q_u \,. \tag{29}$$

The forces F_u with some additional damping are included in the rigid body dynamics equations of motion. Using this methodology, a finite element model of the lumbar spine was incorporated in the occupant model, and the loads acting on the spine were calculated. Preliminary results show that the new spinal model can work successfully as compared to experimental data. Further investigations are being conducted in order to construct a layered model with alternate discs and vertebrae.

5.3 DUPLICATION OF EXPERIMENTS

Some experimental studies, including deceleration sled tests with forward-facing and pitch configurations were performed at Civil AeroMedical Institute (CAMI) [13,14] and National Institute for Aviation Research (NIAR) to validate the SOM-LA/TA model. A sample of the results from the CAMI tests and those of the simulations is shown in Figure 16. The figure shows the time history of pelvis resultant acceleration of a 50th percentile male dummy (HYBRID II) seated over a general aviation seat in a forward-facing test. The input deceleration pulse had a trapezoidal shape with 12 g amplitude and 150ms duration. The sled velocity right before impact was approximately 14 m/s. As shown, the results from the computer simulation match the experimental results well.

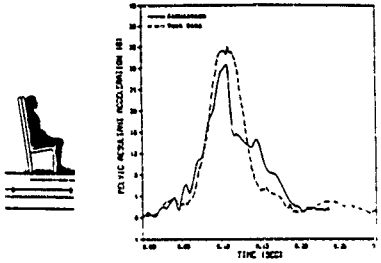


Figure 16. Dummy pelvis resultant acceleration versus time.

Some tests have also been performed at the Impact Dynamic Laboratory of NIAR and have been duplicated using the developed analysis capabilities. As an example, Figure 17 shows the resultant head acceleration of a HYBRID II dummy versus time. The deceleration pulse corresponds to Federal Aviation Regulations Part 23_Test_1 for a passenger, which is a 15g triangular pulse with 120ms duration. The dummy was restrained using a lap belt to a rigid seat at 60 degree pitch configuration. As observed, the results from both analysis experiment match closely.

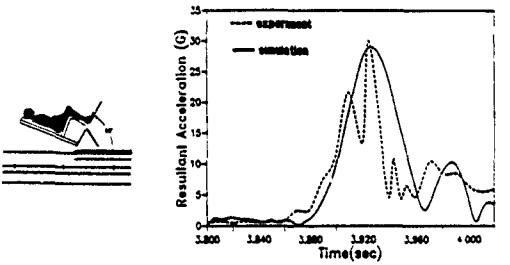


Figure 17. Dummy head resultant acceleration versus time.

5.4 HEAD INJURY CRITERIA

It is well recognized that a head impact produces both translational and rotational motion as well as deformation of the skull. Resultant brain injury may occur from both absolute motion of the brain and its relative displacement with respect to the skull. At present, there are several physical parameters used in the evaluation of head injury, including translational and/or rotational acceleration levels of head impact, impact force, velocity and kinetic energy, impulse and impact duration, etc. These measures have been widely used for animal, human cadaver, and dummy experimental data to determine tolerable and survival thresholds for head impact in translation or rotation. Other parameters such as skull displacement and stresses, brain pressures and strains, as well as neck stretch/strain are usually related to analytical and experimental head model studies.

Head Injury Criteria (HIC) is generally accepted as an indicator of the likelihood of severe head injury and is determined from [25]

$$HIC = \left\{ (t_2 - t_1) \left[\frac{1}{(t_2 - t_1)} \int_{t_1}^{t_2} a(t)dt \right]^{2.5} \right\}_{max} \qquad (30)$$

where t_1 and t_2 are the initial and final integration times, respectively, and a(t) is the resultant head acceleration (g) verses time (s) curve for the head strike. The HIC is a method for defining an acceptable limit; i.e., the maximum value of the HIC should not exceed 1000. It is part of the Federal Motor Vehicle Safety Standards (FMVSS) and Federal Aviation Regulations [24]. Program SOM-LA/TA computes the HIC by a moving-window integration of the data and a specific choice of data sampling.

5.5 CONTACT FORCE MODEL

To assess our capabilities in predicting HIC during a head impact with the bulkheads of the aircraft, several analytical simulations were performed and the results were compared with those of a dynamic test carried out at CAMI. The protocol in the experiment was developed to measure the head path and velocity from an anthropomorphic test dummy (ATD) restrained in a passenger seat. A mockup of a vertical wall in front of the seat was included in the fixtures. The distance between the seat end and the bulkhead was 0.9 meters. The test setup is shown in Figure 18. The input pulse was of a triangular shape with a maximum value of 16.9 g and a span of 160 ms. The velocity of the sled at the time of impact was 14 m/s. A rigid seat was selected in order to eliminate the effects of the seat response on the occupant performance. The rigid seat consisted of two planes that represent the seat pan and the seat back. Cushions were included on the top of the seat pan and seat back surfaces. The seat pan remained fixed in the aircraft floor and the seat back was permitted to rotate forward about a transverse hinge axis at the base of the back if pushed from behind. A standard 50th percentile dummy (FAA required) was then chosen as the occupant.

Series of analytical studies were then performed to match the results from the CAMI test. Figure 19 shows the head strike impact from simulation at different frames. In order to assess the compliance characteristics and displacement requirement of the aircraft bulkheads that minimize injuries as a result of a head contact, an envelope for the occupant and seat was generated, and a more accurate contact force model was also developed.

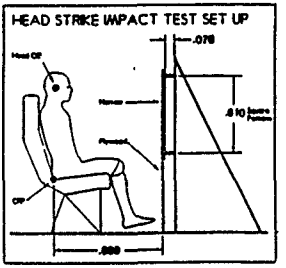


Figure 18. Test set-up of a vertical wall in front of the seat

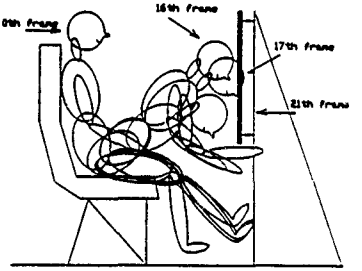


Figure 19. Head strike impact from simulation at different frames.

To match the experimental results, a nonlinear, viscoelastic-type contact force model was used such that the contact force f was calculated from the deformation (indentation) δ and deformation rate $\dot{\delta}$ according to

$$f = A(e^{B\delta} - 1) + C\dot{\delta} \qquad (31)$$

for which A and B are the stiffness coefficients, and C is a damping coefficient. Other models including linear viscoelastic, Hertzian with damping, and Hertzian with permanent indentation were also considered. However, the contact force model of equation (31) correlated best with the experimental static tests done on different padding materials For different materials, based on the static tests, coefficients A and B are

evaluated from experimental correlations. For the damping force, the damping ratio is taken to be constant for all the deformation modes of interest. This reduces the Raleigh formulation of the damping coefficient to

$$C = 2k\alpha \tag{32}$$

where k is the gradient of the contact force-deflection curve; i.e., $k = ABe^{Bx}$. Variable α is a representative constant for the system, and is evaluated as

$$\alpha = \frac{C_0}{2K_0} = \frac{C_0}{2AB} \tag{33}$$

where C_0 and k_0 are respectively the damping and gradient of stiffness coefficients for zero deflection condition.

The contact force model was then used to determine the occupant response as a result of a head contact with the wall. The coefficients were varied in order to find out how the changes would affect the HIC and maximum deformation of the front panel. For values of the coefficients A = 3203 N, B = 27.95 1/m, C_0 = 175 N.s/m, the analytical results matched the experimental results from CAMI test. The contact force model corresponding to these values is shown in Figure 20, for both the stiffness part and the damping part. Variation of different coefficients produced some interesting results. When stiffness coefficients A and B are increased, HIC increases linearly, while maximum deformation of the front panel decreases. However, when damping coefficient C_0 is increased, HIC decreases first, and then increases after a while (Figure 21).

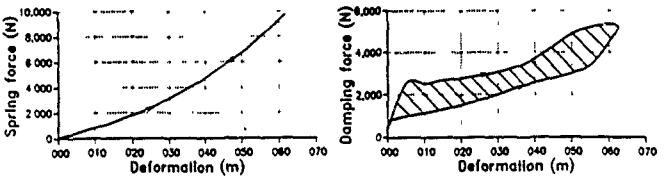


Figure 20 . Contact force models: (a) stiffness part; (b) damping part.

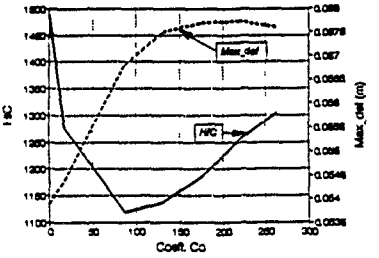


Figure 21. Variations of coefficient C_0.

This means that for a given value of A and B, there is a value of C_0 that minimizes HIC. In order to keep the head injuries minimum, the value of C_0 should be 87.5 N.s/m and the material for the bulkhead, including both frame and padding material, should have about 0.057 meter of maximum displacement requirement.

The parameters A, B and C_0 in the contact force model were then optimized by subjecting the bulkhead deformation δ to be less than 0.057 meters and then minimizing the bulkhead acceleration $|\ddot{\delta}|$. Following the procedures of section 4, the optimized parameters are A= 61.7 N, B=2.8 1/m and C_0= 491.32 N.s/m. Figure 22 shows the variation of contact forces at the optimum stiffness and damping properties. Materials are being searched now, that have similar amount of energy dissipation capability.

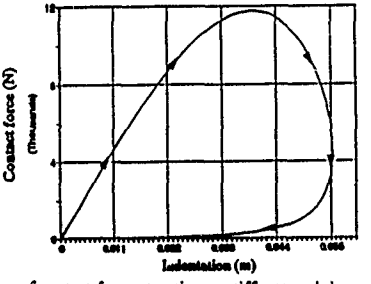


Figure 22. Variation of contact force at optimum stiffness and damping properties.

Based on the preceding discussions, a drop tower was designed to simulate the head strike on the panel (Figure 23). This facility consists of a dummy head assembly, holding assembly, a wire rope guided system, and an impact absorber (panel). The velocity of the system depends on the height from which the dummy head is allowed to fall freely due to gravity. The acceleration is determined by fixing accelerometers inside the dummy head. Two wire ropes, controlled by motors, are used to raise the head assembly to the required height from the ground level. The attainable velocity from this method varies from 5.2 m/s to 17.4 m/s. For a given velocity, the head is dropped onto the absorber, and the amount of crush left on the absorber is recorded. For example, to attain an impact velocity of 14 m/s, the head is released from a height of 10 m. Based on the earlier discussions, an absorber that has a crush of less than 5.7 cm is an acceptable one. For justification of the results, the HIC is also evaluated from the accelerometer data, which has to be less than a value of 1000.

6. Conclusion

A continuous analysis method has been presented for the direct-central impact of two solids. To represent the variation of the contact force during the contact period, two Hertzian models are used, one with hysteresis damping and one with local plasticity effects. At low impact velocities, energy is dissipated in the form of internal damping or heat. If the initial indentation velocity is not negligible compared with the propagation speed of deformation waves across the solid, then permanent indentation is the dominant

factor accounting for energy dissipation. For both models, based on energy and momentum considerations, the unknown parameters were evaluated in terms of the geometrical and material properties of the contact surfaces, velocities of the solids before impact, and the coefficient of restitution. A procedure for impact analysis of multibody systems was then developed using the concept of effective mass. From the comparison of the continuous analysis method with a canonical set of impulse/momentum equations, it is observed that when significant changes occur in a system configuration before and after impact, the use of piecewise analysis method is not suitable for prediction of impact responses.

Use of impact sled tests is the most common technique for determining the post-crash dynamic behavior of an aircraft occupant. The impact sled and target tracking facilities available at National Institute for Aviation Research (NIAR) were used to conduct a study on occupant responses in a crash environment. Parallel analysis capabilities, including a multibody dynamic model of the occupant and a finite element model of seat structures, have been developed. The analysis has been used to reasonably predict the Head Injury Criteria (HIC) as compared with the experimental impact sled tests for an occupant head impacting a panel. A nonlinear viscoelastic contact force model was developed. Suitable values of the coefficients in the contact force model were obtained and the correlations between the coefficients, HIC, and maximum deformation of the front panel were determined. A non-sled test method has been designed to determine the head injuries as well as the performance of a particular impact absorber.

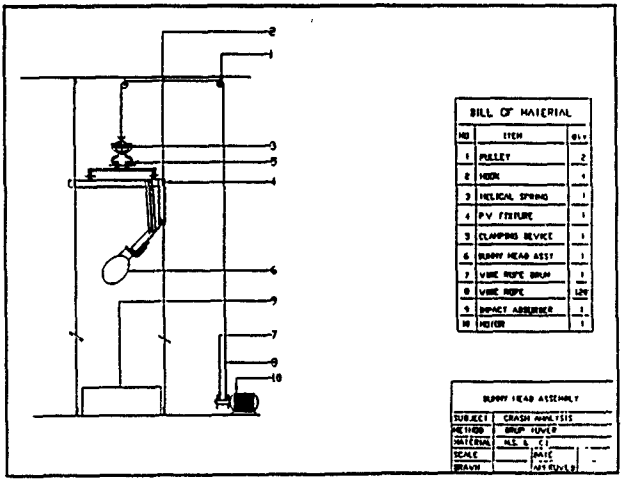


Figure 23. A head-strike test rig.

This research is partially funded by U S Department of Transportation, Federal Aviation Administration Technical Center, Atlantic City Airport, New Jersey. It is also partially funded by Wesley Medical Foundation, Wichita, Kansas.

7. References

1. Greenwood, D.T., **Principles of Dynamics**, Prentice-Hall, Angled Cliffs, NJ, 1965.
2. Meirovitch, L., **Methods of Analytical Dynamics**, McGraw-Hill, New York, 1970
3. Kane, T. R., **Dynamics**, Holt, Rinehart & Winston, New York, 1968.
4. Wehage, R. A., "Generalized Coordinate Partioning in Dynamic Analysis of Mechanical Systems," Ph.D. Dissertation, University of Iowa, Iowa City, 1980.
5. Khulief, Y. A. and Shabana, A. A., "Dynamic Analysis of Constrained System of Rigid and Flexible Bodies with Intermittent Motion," *ASME Journal of Mechanisms, Transmissions, and Automations in Design* Vol. 108, pp. 38-44, 1986.
6. Lankarani, H. M. and Nikravesh, P. E., "Canonical Impulse-Momentum Equations for the Impact Analysis of Multibody Systems," *ASME Journal of Mechanical Design*, Vol. 114, pp. 180-186, 1992.
7. Goldsmith, W., **Impact, the Theory and Physical Behavior of Colliding Solids**, E. Arnold Ltd., London, 1960.
8. Khulief, Y. A. and Shabana, A. A., "A Continuous Force Model for the Impact Analysis of Flexible Multibody Systems," *Mechanism and Machine Theory*, Vol.22, PP. 213-224, 1987.
9. Hertz, H., **Gesammelte Werke**, Vol. 1, Lepzig, Germany, 1895.
10. Hunt, K. H. and Grossley, F. R. E., "Coefficient of Restirution Interpreted as Damping in Vibroimpact," *ASME Journal of Applied Mechanics* Vol. 7, PP. 440-445, 1975.
11. Crook, A. W., "A Study of Some Impacts Between Metallic Bodies by a Piezoelectric Method," Proceedings of the Royal Society London A212, PP. 377-390, 1952.
12. Bernhart, K. E., "Transverse Impact on Elastically Supported Beams," Ph.D. Dissertation, University of California, Berkeley, 1955.
13. Laananen, D. H., Bolukbashi, A. O., and Coltman, J. W., "Computer Simulation of an Aircraft Seat and Occupant in a Crash Environment, Vol I - Technical Report, TR-82401," Simula Inc., Tempe, Arizona; DOT/FAA/CT-82/33-I, U.S Department of Transportation, Federal Aviation Administration Technical Center, Atlantic City Airport, New Jersey, September, 1982.
14. Laananen, D. H., "Computer Simulation of an Aircraft Seat and Occupant in a Crash Environment - Program SOM-LA/SOM-TA User Manual," DOT/FAA/CT-90/4, U.S Deparument of Transportation, Federal Aviation Administration Technical Center, Atlantic City Airport, New Jersey, May, 1991.
15. Lankarani, H. M. and Nikravesh, P. E., "Continuous Contact Force Models for Impact Analysis in Multibody System," Proceedings 4th Conference on Nonlinear Vibrations, Stability, and Dynamics of Structures, Virginia Polytechnic Institute,1992, *Journal of Nonlinear Dynamics*, Kluwer Academic Publishers, 1993.
16. Lankarani, H. M. and Nikravesh, P. E., "A Contact Force Model with Hysteresis Damping for Impact Analysis of Multibody Systems," *ASME Journal of Mechanical Design*, Vol. 112, pp. 369-376, 1990.
17. Love, A. E. H., **A Treatise on the Mathematical Theory of Elasticity**, 4th ed., Dover Publications, New York, 1944
18. Lankarani, H. M. and Nikravesh, P. E., "A Hertz Contact Force Model with Permanent Indentation in Impact Analysis of Solids," ASME Advances in Design Automation, Design Technical Conference, Scottsdale, 1992.

19 Lankarani, H M., "Canonical equations of Motion and Estimation of Parameters in the Analysis of Impact Problems," Ph D. Dissertation, University of Arizona, Tucson, 1988
20 Gim, G., Pereira, M.S., Lankarani, H M. and Nikravesh, P. E . "Technical Data and Plastic Hinge Model for M151-A2 Rollbar Model," Tech. Rep. No. CAEL-87-4, Univ. of Arizona, 1987.
21 Gim, G., Pereira, M.S., Lankarani, H. M. and Nikravesh, P. E . "Rollover Analysis of Vehicles with Safety Rollbars," Tech. Rep. No CAEL-87-5, Univ. of Arizona, 1987.
22. Ma, D., Periannan, K., and Lankarani, H.M ," Occupant Dynamic Responses in a Crash Environment," Proceedings 4th Conference on Aircraft Interiors and Safety, National Institute for Aviation Research, 1992.
23. Merrill, T., Goldsmith,W., and Deng, Y C., "Three Dimensional Lumped Parameter Head-Neck Model due to Impact and Impulsive Loading," *Journal of Biomechanics*, ol 17, No. 2, pp.81-95, 1984.
24. Steele, B., and Lankarani, H.M., "A Brief Examination and Comparison Between the Federal Motor Safety Standards and the Federal Aviation Regulations," National Institute for Aviation Research Tech. Rep No 91-25, Wichita State University, 1991
25 Malapati, R., and Lankarani, H.M., "Evaluation of Head Injury Criteria," National Institute for Aviation Research Tech. Rep. No. 92-20, Wichita State University, 1992.

DATA FITTING METHODOLOGY FOR FRONTAL CRASH VICTIM SIMULATION

Christian Goualou - Eric Vittecoq - Jean-Pierre Faidy
L R S A. Research Laboratory for Automotive Safety
Ecole Centrale Paris - P.S.A. Peugeot Citroën
Châtenay-Malabry, France

ABSTRACT. The aim of the paper is to present the methodology for data fitting we developed and followed to perform crash victim simulations. The project carried out with P.S.A. Peugeot Citroën car company, consisted in modelling and fitting efficiently all interactions between a dummy and a car during a frontal crash. This work was accomplished on the basis of standard crash experimental measurements without making systematic ticklish test identification on parts of the system. The final purpose is to optimise the safety equipments. For this study, we choose to consider the dummy as a rigid multibody system and to use the program MADYMO from T.N.O. (Delft)
- The paper will first briefly present the system to model ;
- Then, input and output parameters will be identified and classified. The methodology used to fit the model to sled tests will be presented, with the help of a sensitivity analysis and/or a rules system. We will explain how this approach permitted us to correlate the model for very different crash test configurations with the same basic data. Partial results of two simulations will be shown ;
- Finally, it will be explained how to take into account in the simulation effects of a few phenomena which occur in a real crash and are not described by the theoretical model.

1. Introduction

Numerical methods have been recently introduced for the simulation of various configuration automotive crash tests. In this field of mechanical analysis, our aim is to evaluate the coupling between the car and the dummy, during a frontal crash. By using a relatively simple model, the principal purposes of computation can be : Understanding during the modelling and evaluating the fitted system by simulation. Our paper deals with this first aspect.

1.1. UNDERSTANDING DURING THE MODELLING

- To know what happens qualitatively in a very short time (0.2 second) in terms of the succession of contact, deformation (springs) and motion of different parts of the system,
- to understand the behaviour of the dummy and the restraint system, meaning the signification and the influence of few global parameters.

1.2 EVALUATING THE FITTED SYSTEM BY SIMULATION

- In order to compare different configurations of crashes using a change of one or several significant parameter(s) without making costly experiments ; and with the insurance that, excepted for wanted changes, all parameters remain unchanged,
- and to evaluate the hierarchy of the parameters and their interaction,
- to show tendency and sensitivity of the system by governing parameters,

- and to finally opumise the system.

1 3 A GENERAL PROBLEM FOR THE MODELLING

In fact beyond our practical project, we were aware of a general problem during the building of a numerical simulation model. A lot of standard methods and programs are available today. The engineer, after having formulated his theoretical model, needs to find the best input data set for a specific application, fitting closely the system. For that purpose, he may analyse the experiment data and conditions to find some parameters. In fact, all input data is not easily identifiable during the experiment (even the geometry). A common approach is to make systematic test identification on parts of the system, especially for stiffnesses. But often, such a partial test is very difficult to conduct, and it provides data that doesn't fit well the crash reality or/and doesn't lead to a good simulation. As a consequence, it becomes necessary to modify the test data to obtain a better correlation. With this problem in mind, we choose to minimise the number of such tests, to make the more practical and reliable of them (example : static retractor or seat pan stiffnesses) and to manage the fitting of ticklish parameters by finding the best correlation (example : friction coefficient). Even though we knew that the more the system is coupled, the more difficult is the formulation of the data, we were sure that this approach will help us to understand better the phenomenon. This approach justified the motivation in choosing a relatively simple model, with the advantage of a very short computing time (a few minutes on a Unix station).

2. The system to model

The interaction between the car and the dummy is our principal interest. Thus the model concerns only the dummy and its direct environment (figure 1). Even if we began first to consider sled tests, after having noticed rather 3D motion for the dummy in car crashes, we choose to built directly a 3D model.

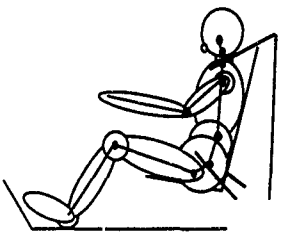


fig 1. The dummy and its environment

The dummy is modelled by 13 rigid parts (materialised by the skeleton on the sketch) which are connected by 12 joints. Each joints are spherical and owns complex angular loading and unloading characteristics. A certain number of ellipsoids are added to the members, on which a contact may occur. The model allows contact between ellipsoids and ellipsoids or planes, typically between pelvis and seat, foot and foot planes, chin and upper torso, arms and legs, head and knees, etc.
In reality, there is a deformation when a contact occurs. The model detects a simple geometric interference and generates a complex force function of the interference and its velocity. For each force-deflexion law in the model (joint or contact), it is possible to introduce friction, damping and hysteresis.

As can be seen in the figure 1, for the environment we incorporated a seat, a standard 3 points seat belt, a foot pan and a toe board. It should be noticed that for a real car, it will be essential to model the dash board.

The belt may be decomposed in 5 segments, whose extremities are on the car's components or on the surface of the dummy. Each segment acts like a linear spring and owns force-deformation characteristic. Between two adjacent segments, on the common (buckle with null diameter) or distanced points (contact on the dummy), there is a close relation given by the classical law of friction (belt and pulley) and then a possibility for the strap to slip along its route formed by the different connected (so fixed) points. A retractor (film spool effect) may be simulated by correcting the stiffness of the corresponding segment.

The analysis supposes that the car (or sled) is fixed to the reference frame and the dummy is accelerated forward by the opposite of the acceleration-time history. This history is measured and filtered on the middle pillar of the car during a real crash or sled test, see figure 9. We are working with a relative motion between the car and the dummy : Such a description involves the neglecting of the car rotation (coriolis effect). To obtain the total acceleration on the dummy parts, the calculated acceleration and the prescribed one have to be added.

All this aspects were commonly available in the program Madymo [1] and have not needed specific implementation.

3. Data fitting

3 1. DATA SET AND INPUT PARAMETERS

The dynamic description of the system evoked above, includes inertia, joints and contact properties, which correspond to so-called global parameters (a single value or a function). These parameters may be classified in 2 types : Intrinsic (I), and configuration Test dependent (T). By intrinsic, we mean that these parameters have to be reproductible with one single level. This level is fixed by a certain technology, we don't want to change, dummy properties or friction coefficient for example. Futhermore, another distinction may be applied : Validated (V) (from reliable partial experiment or precedent test correlations), Directly identifiable (D) (from the test) and Fuzzy (F) at the moment.

A first class of parameters is available at the beginning :

- dummy properties	I	V
- dummy installation (geometry)	T	D
- dummy environment (geometry)	T	D

A second class of parameters that have to be fitted is given with their hierarchy :

Main parameters :		
- belt slack and toe board - foot slack	T	F
- belt segments stiffness	I	F
Minor parameters :		
- seat and foot rest stiffness	I	F
- retractor characteristic	I	V
- friction coefficient between strap and buckles	I	F
- friction coefficient between strap and dummy	I	F
- friction coefficient for the various contacts	I	F

Note that for our simple model, where the dummy is modelled as rigid, the unknown flexibility of the upper torso and pelvis is incorporated in the belt segment characteristics Thus, it is necessary to modify deeply the common static force-deformation for the strap (9% of deformation under 10 kN).

3.1 1. *Intrinsic parameters.* Dummy data : With Madymo, the model was almost ready to be used. A data base for the dummy, we employed for the sled test, namely the Part 572 dummy, were furnished with the code. It contains all the validated dynamic properties of the dummy. We found useful to modify the ellipsoids and the location of the accelerometers proposed in the previous data base.

The final purpose is surely to determine precisely the other intrinsic and often fuzzy parameters, after what they will become validated, but a with good portability of them on several tests. For the very first test, it is also necessary to estimate them rougly using mechanical sense and experience.

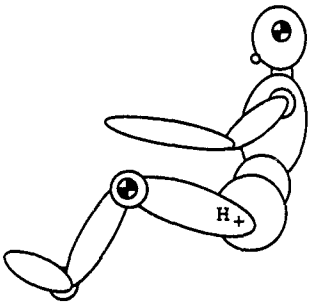


fig 2. Two targets to determine the position of the hindered joints

3.1.2. *Test dependent parameters.* Dummy installation : The dummy, our principal interest, is taken as the centre of the geometric world of analysis. It is not a simple task to place it in a accurate position (2D problem), before adding the other elements around it. As the dummy is formed by rigid bodies linked by spherical joints, we must determine the orientation of them and choose a reference point. The main difficulty concerns the localisation of the hindered joints for the spine. A geometrical method has been used assuming that there is a zero torque initially in the spine's joints. As a consequence, the spine may be considered frozen and one degree of freedom exists for the dummy position, which is the relative angle between the global torso and the upper legs.

With the position of two targets, respectively on the head and knee, given by the film, it is very simple to deduce analytically the H point (centre of the pelvis-upper legs joint) and the whole position of the dummy torso and upper legs. The other members orientations are identified from the film easily.

Belt route : Another problem is to find the routes of the shoulder and lap belts on the dummy, with the evidence that this routes often may change during a crash, corresponding to transverse slip not modelled in Madymo. Another reason to treat this problem analytically is that we wanted to simulate earlier tests, where the connecting points of the strap on the dummy were not known. Therefore we choose to model the route and calculate geometrical characteristics of the belt with a pre-processor, in the initial position of the dummy. It is explained below how to control this parameter.

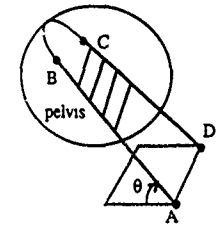


fig 3. One single parameter θ to find the route of the lap belt

The corresponding model is built by assuming that the shoulder belt, as the lap belt (example in figure 3), were each contained in a plane and that each segment is tangent to the ellipsoid to be considered. As the plane must be built with a line formed by the two anchorage points (A and D), it may be driven by a single parameter as an angle around this line. For such a parameter, the pre-processor provides the connecting

points coordinates on the torso, respectively the pelvis (B and C), and the length of the strap in contact. It is interesting to notice that during the simulation of various tests, the maximum corresponding plane default doesn't exceed 3 cm for the shoulder belt.

Other elements : According to certain chosen geometric parameters (initial equilibrium), our pre-processor computes the whole coordinates for the seat and the foot planes.

3 2. OUTr JT PARAMETERS

Output parameters are commonly recorded as time histories during a test. They may be classified for the data fitting :

Main parameters :
- upper torso backward acceleration
- belt segments forces
- targets trajectories (head, shoulder, pelvis and knee)

Minor parameters :
- the other accelerations (components, for the head, upper torso and pelvis)

Ticklish parameters :
- head acceleration components
- head trajectory (mainly at the end of the crash)
- shoulder trajectory

3.3. METHODOLOGY

The ease of data fitting is conditioned by the knowledge we have initially about the common input parameters. We shall explain how we began this study in an complete unknown. The idea is to make a first rough fitting and to proceed after by refinements. Obviously, after having determined the intrinsic parameters on several correlations, it will be easier and faster to simulate other tests.

3.3.1. *First rough fitting (first simulation of a first test).* All the parameters are initially estimated.

The first step consists in correlating the upper torso backward acceleration, because it is in a close relation with the initial kinematics energy of the dummy and so mainly affected by only 2 main parameters. Thus, first stage : the starting time of the acceleration curve gives the global slack of the belt (Test,dependant). Second stage : the stiffness of the shoulder belt (Intrinsic) is fitted by correlating in timing and height, the maximum acceleration (and the maximum force in shoulder belt segments if measured).

The second step consists in correlating knee and pelvis targets trajectories. Slacks (T) and stiffness (I) of the belt lap and the toe board are determined by the horizontal excursions : and stiffness (T) of the seat is determined by the pelvis penetration. The friction coefficient (I) between the seat and the pelvis is adjusted in correlating the maximum of lap belt force.

In a third step, we may fit the friction coefficients (I) concerning the belt, by correlating the various force-time histories of the belt segments and the head target trajectory.

For next correlations of other tests, we shouldn't modify the intrinsic and work out only with the test dependant parameters. If a better correlation is obtained by changing an intrinsic parameter, we have to find an explanation of this, and examine the effect of this potential change on the previous simulations.

3.3.2. *Refinement.* As the parameters are strongly coupled, it is not evident to find a good solution for the input data set. In changing one parameter, the whole output parameters

may be affected, conducting to review the other input parameters previously fitted... Thus it is important to proceed regarding the hierarchy of the parameters. After a first rough correlation, it becomes necessary to play with all the parameters. In these conditions, only the experience of the engineer (or simple ticklish identification test) may permit to reach an acceptable solution. Thus we choose to make a sensitivity analysis, which is presented figure 5 in a qualitative matrix on one hand and to heap up observed effects of changing parameters in a formal rules system on the another hand.

For the building of this matrix, we have to make several remarks :

- The inputs (one line) are supposed increased (tuned up from an estimed to a maximum feasible value), and we look at the qualitative variations of the outputs (relative to an experimental scale) : 0 means no influence, + or - a significant one, ++ or - - a strong one and + - a non-monotonic dependancy ;
- We consider here a parameter as a globalised data. For example, at an increase of the belt suffness corresponds an amplification factor on the basic low (force-deflexion) ;
- For the belt angle : an increase of it provides higher connecting points on the dummy ellipsoid (see paragraph 3.1.2.) ;
- The f notation is an abbreviation of friction coefficient.

The sensitivity matrix, regarding certain output parameters, is very useful in predicting which input parameters have to be increased or decreased to obtain a better correlation. However the validity of the matrix keeps staying local and it is therefore necessary to furnish an initial approximation of the data set not too far from the solution and to update the matrix when getting a good experience of the system behaviour.

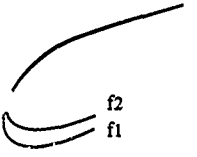


figure 4. example of a rule

As the rules system contains a lot of relations, it is not easy to present it in this paper and we give here only an example, figure 4, between friction coefficient dummy-strap and head trajectory : an increase of the friction coefficient (f_1 to f_2) leads to a reduction of the vertical component of the side view trajectory of the head.

This tools, implemented at each new test correlation, permitted quickly to describe above numerical results, phenomenological behaviour and to improve the data set, with common intrinsic parameters or with identified differences.

4. Limits of the model

4.1. INTRODUCTION

Several problems occur during the data fitting approach. Some phenomena, that can't be evaluated experimentally for a common test, seem to influence sharply the behaviour of the dummy, and are beyond the capabilities of the model. For example, large rotation of the pelvis may introduce a global stiffness for the contact with the seat, very different than a previous measurement with the pelvis normally seated. Ever for this contact, the standard friction model, which at a zero relative speed, could only correspond a zero friction force, leads to unrealistic acceleration for the pelvis when it begins to come back, even unstability. It is beyond the scope of this article to explain all the numerous problems of modelling, we had to deal with, but the most important, that are the pillar(1)-ring clamping and the torso twisting control.

Inputs \ Outputs		Accelerations				Trajectories						Belt forces								
		starting time (backward)	max. (upper torso)	max. resultant head	max. resultant pelvis	head (amplitude)	left shoulder (amplitude)	pelvis or knee (amplitude)	head (return)	pelvis (amplitude)	torso twisting (120ms)	starting time	max. retractor part	max. upper (shoulder belt)	max. lower (shoulder belt)	max. inboard (lap belt)	max. outboard (lap belt)	timing for return to zero	max. normal reaction of the seat
Belt	global slack	++	++	+	+	:	:	-	-	+	0	++	+	++	++	++	++	+	++
	global stiffness	0	0	0	0	++	-	0	++	-	0	0	0	0	0	0	0	0	-
	hysteresis	0	0	0	0	-	+	0	-	+	0	0	0	0	0	0	0	0	+
	[strap/dummy	0	0	0	0	-	-	0	-	-	0	0	0	0	0	0	0	0	+
	[upper buckle	0	0	0	0	-	-	0	-	-	0	0	-	+	+	+	+	0	+
	[lower buckle	0	0	0	0	-	-	0	-	-	0	0	0	+	+	0	0	0	+
	retractor stiffness	0	+	0	0	-	-	0	++	-	0	0	++	+	++	-	0	;	0
	shoulder belt angle	0	0	+	-	-	++	0	0	0	-	0	-	-	-	-	-	0	++
	lap belt angle	0	+	-	-	0	++	+	0	0	++	0	-	0	-	-	-	-	+
Seat	stiffness	0	-	0	++	+	0	0	0	0	+	0	+	+	+	+	+	-	0
	hysteresis	0	0	0	++	+	0	-	0	0	+	0	0	0	0	0	0	0	0
	[with dummy	0	-	0	++	+	-	-	-	-	0	0	0	0	0	0	0	0	0
Toe board	slack with feet	0	+	-	++	0	-	-	-	-	0	0	+	+	+	+	+	+	++
	stiffness	0	0	0	+	+	0	-	-	0	0	0	0	0	0	0	0	0	-
	hysteresis	0	0	0	0	0	0	÷	-	-	0	0	0	0	0	-	-	0	+
	[with feet	0	0	+	+	+	0	+	0	0	0	0	0	0	0	0	0	0	+

fig 5. A sensitivity analysis (low severity test simulation)

4 2. PILLAR RING CLAMPING

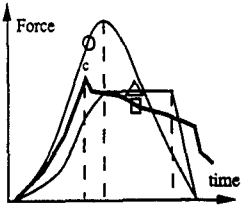


figure 6. Belt forces-time histories

For the model as the retractor part is passive, slip at the pillar ring acts every time, except during the maximum flat level, see figure 6. Thus, a simple friction relation exists between the tensions of the two belt segments : Retractor part and shoulder upper belt. These forces keep growing always in a ratio given by the coefficient friction at the pillar ring and the angle formed by the segments (none constant). As a consequence for the time history : After having caught the slack, the forces increase in this ratio with their maxima reached in the same time. After the maxima the retractor part stands at the same level when the upper part begins to decrease, until the ratio of slip is reached again. Then, the retractor part decreases too.

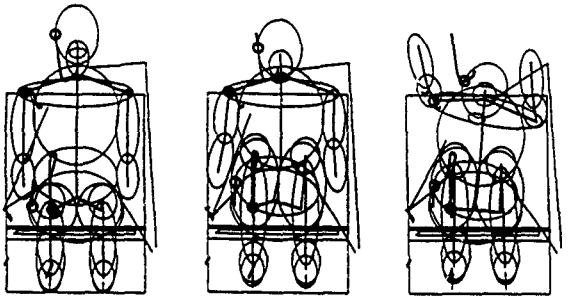
In the reality of a test, the retractor part force-time history doesn't follow the model. It seems to prove that the friction coefficient is very small at the beginning until it increases a lot. In other terms, a clamping occurs (c). The consequences of this phenomenon are severe for the head, for which it becomes impossible to reproduce accurately acceleration and trajectory. In order to take into account of this phenomenon, a high constant friction coefficient has to be used.

As an example : For the second test presented in this paper, this phenomenon occurred surely. In the simulation, the head acceleration is lowered, and the trajectory increased. Of course, it would be possible to correct the simulated trajectory by reducing the belt slack, but it would offset the maximum time and perturb the correlation of the upper torso acceleration too

4 3. TORSO TWISTING CONTROL

In figures 7, 8, two series of sketches of . same test simulation are presented. The only difference concerns the initial route of the shoulder belt on the upper torso. The first simulation reflects a 2D behaviour. For the second, the torso twisting is very pronounced. According to our method to describe the connecting points (with a single angle to fit the initial supposed belt plane), we need to be careful about this parameter and to control it by an agreement of 3D kinematics of the dummy torso. Of course this control must be made during the data fitting process. We must point out that this ticklish phenomenon is in reality controlled by the initial position of the belt on the shoulder and the transverse slip. It could be perhaps well modelled with a different approach (finite elements). However we think that it is vital to understand this global phenomenon.

In the same way, it is necessary to control the lap belt route, in order to get a good correlation for the parameters concerning the pelvis. According to the more or less realistic rotation of this element in the simulation, the connecting points must be placed relatively high in the initial position, so that, at the maximum tension time, the route seems correct

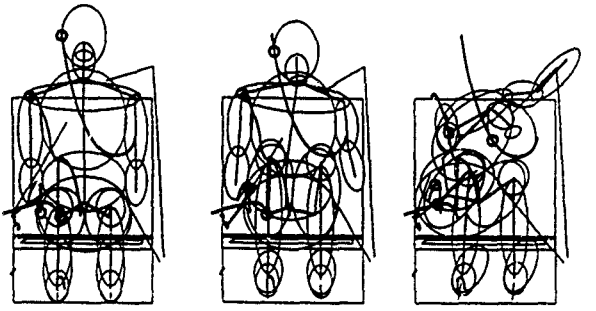


0 ms 60 ms 100 ms

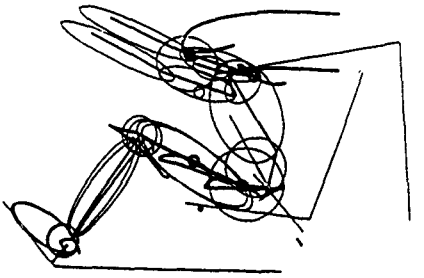
above : near the neck

figure 7 : - influence of the shoulder belt route on the dummy kinematics - front view -

below : middle of the shoulder



0 ms 60 ms 100 ms

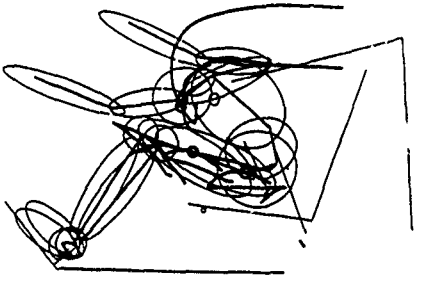


100 ms

above · near the neck

figure 8 : - influence of the shoulder belt route on the dummy kinematics- side view -

below : middle of the shoulder



100 ms

5. Two examples of correlation

As an illustration of this approach, we present here, figures 9 to 16, in the annex, the results of simulations of 2 sled tests, managed with the same intrinsic data and methodology presented in this paper. This tests were realised in very different configuration and may be seen as extremes at the speed of 48 km/h impact (figure 9). The second simulation was so severe, that it was useful to reinforce the seat at his front, and a rigid rod may be observed in a simulation (which interacts with the pelvis). For this test, we think that a Pillar-ring clamping occurs, with the difficulty to correlate both acceleration and trajectory of the head.

6. Conclusion

Thanks to its flexibility, good reliability and low computation time, numerical multibody simulation can be very useful in understanding the global behaviour of a complex mechanical system and further in improving it. However, it's often a long way to fit accurately a formulated model with the experiment. What we attempted to do primarily in this project, was to make a systemanc approach in correlating such a model, in the field of automotive safety. We thought that only a formulation of consistent data will permit to simulate different configuration tests and interpret the influence of global parameters, before performing more accurate but complex computation with finite elements method, for example.

7. Acknowledgements

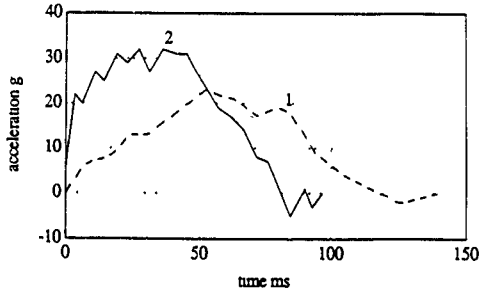
The authors would like to thank G. Brutel, J. Hamon of P.S.A. Peugeot Citroen and D. Aubry, J.L. Burgaud, J.C. Bocquet of Ecole Centrale Paris, for their contributions

8. Reference

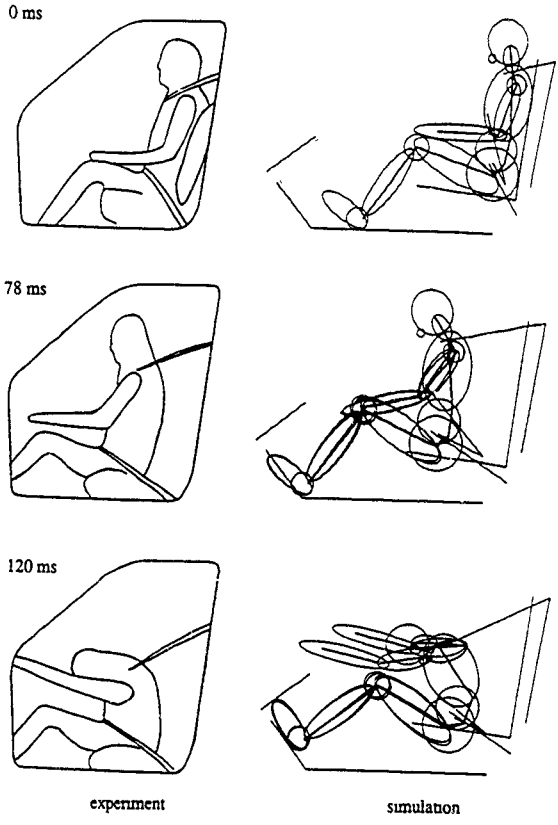
[1] TNO Road-Vehicles Research Institute, Departement of Injury Prevention : Madymo User's Manuel 3D, version 4.3, the Netherlands, 1990

ANNEX : **Results of two simulations**

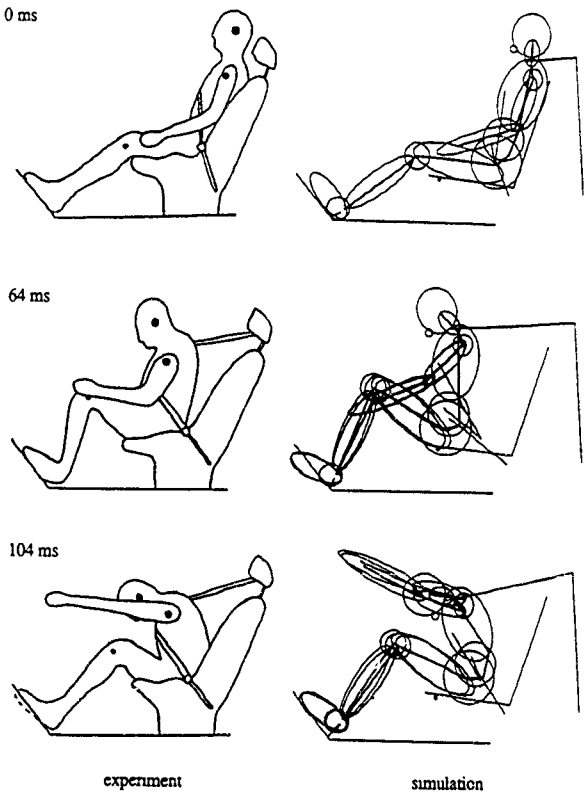
Remark : for all histories, the time unit is the ms.



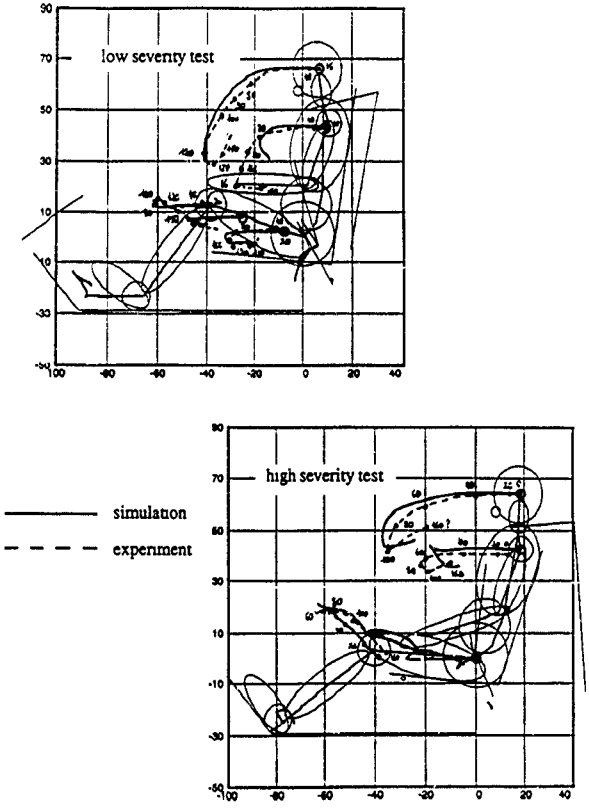
- figure 9. sled acceleration-time histories -
test 1 : low severity and test 2 : high severity



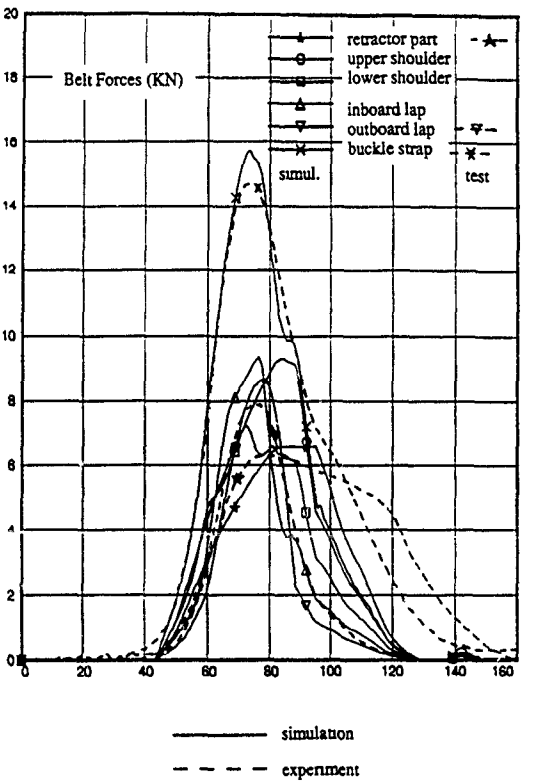
- figure 10. Kinematics for the low severity test -



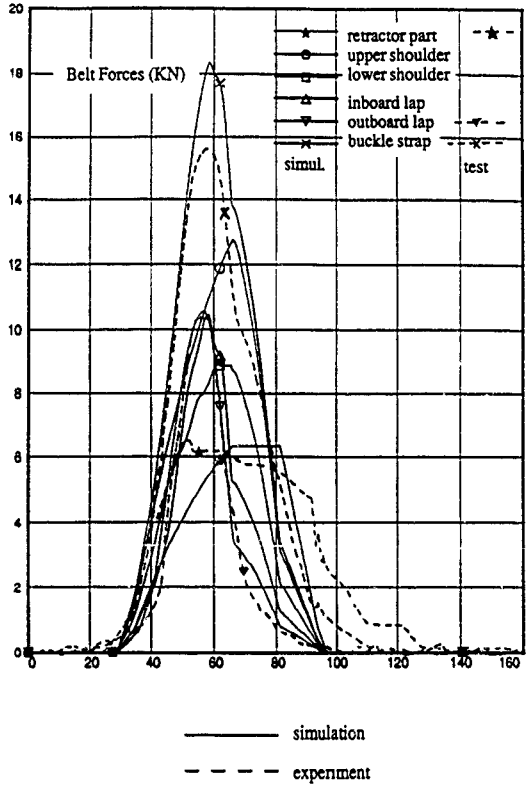
- figure 11. Kinematics for the high severity test -



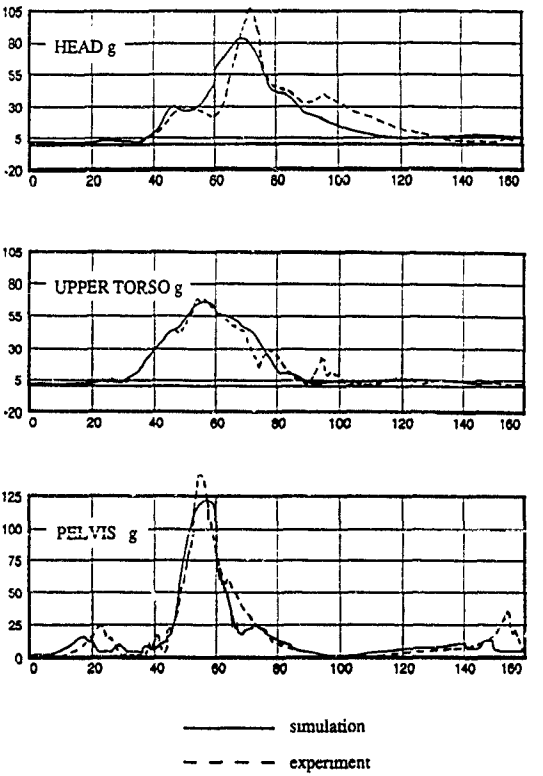
- figure 12. Compared trajectories -



- figure 13. compared time-belt-forces histories for the low severity test -

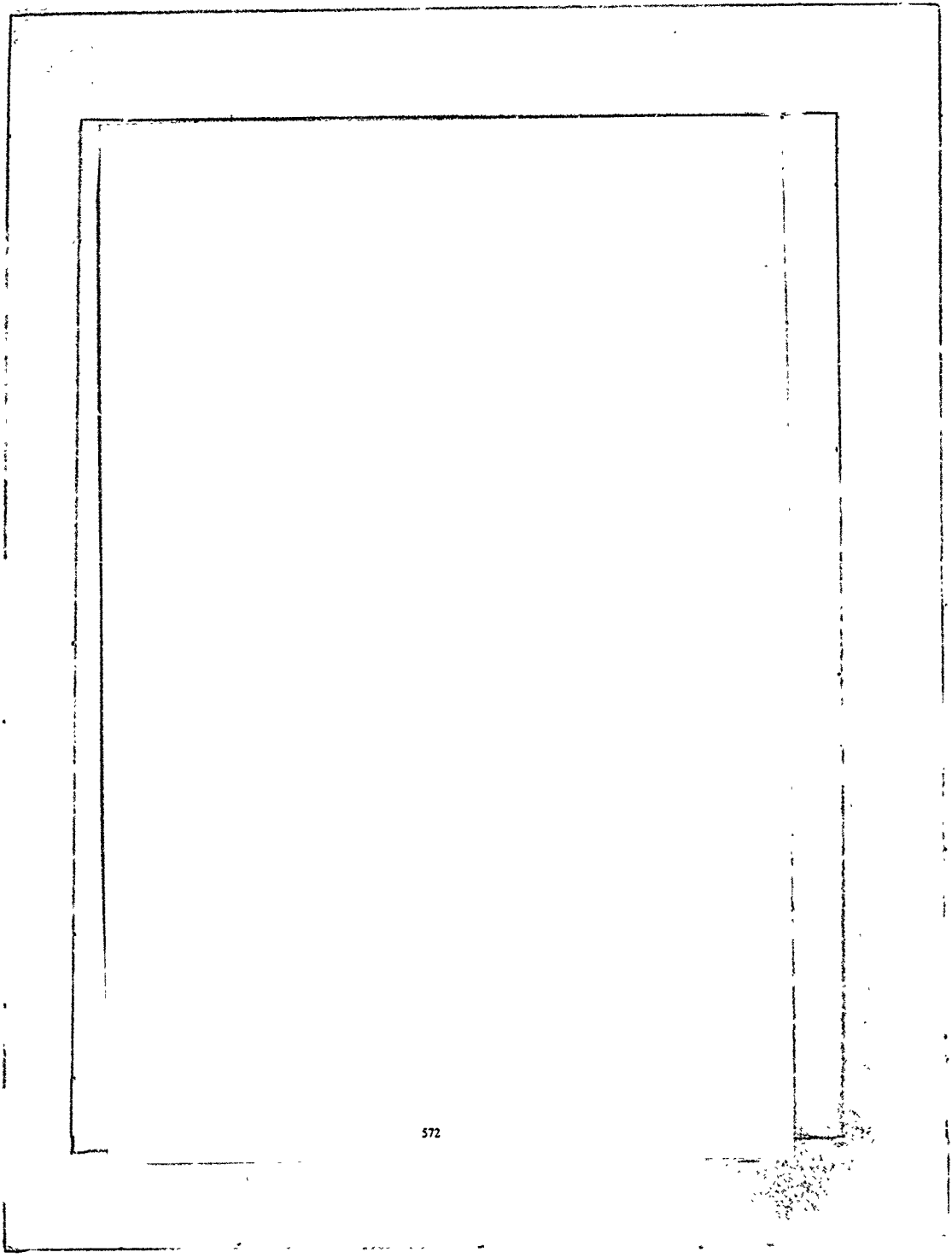


- figure 14. compared time-belt-tensions histories for the high severity test -



standard location of accelerometers

- figure 16. Time-resultant acceleranons histories for the high seventy test -



ANALYSIS AND DESIGN FOR VEHICLE CRASHWORTHINESS

JOÃO M. P. DIAS and MANUEL SEABRA PEREIRA
IDMEC - Instituto de Mecânica
Instituto Superior Técnico
Av. Rovisco Pais, 1096, Lisboa CODEX
PORTUGAL

ABSTRACT. This paper describes a methodology for the sensitivity analysis and optimization of planar constrained mechanical systems. Direct differentiation methods and finite difference techniques have been used in the design sensitivity calculations. The optimization process is developed within the framework of mathematical programming techniques. The sensitivity equations were constructed symbolically and subsequently integrated in the dynamic analysis equations of motion and solved simultaneously.

Some examples are solved to demonstrate the use of the methodology including a crashworthiness design problem, where a plastic hinge concept has been used to model plastic deformations of the vehicle structure during the impact.

1. Introduction

During the last twenty years, computer aided analysis of crashworthiness and structural impact has received a large attention and is about to emerge as a powerful methodology to be successfully applied in practical situations.

Different numerical formulations with varying degree of complexity and accuracy have been proposed by Kamal [1] using spring mass models and by Pifko and Winter [2] and Hayduk et al. [3], using finite element methods. Recent efforts in the field of non-linear structural dynamics have contributed for the development of well known commercially available codes such as PAM-CRASH [4] and RADIUS which are enhancements of the public domain codes DYNA-3D [5] and WHAMS [6]. These programs can simulate with improved accuracy several different structural impact phenomena, such as large localized deformation, structural instabilities, transient vibrations and stress wave propagation. These codes, however, require large computer resources which make them rather unsuitable as part of a design tool to be used in the early design stages.

As a result, design for crashworthiness is, in most industrial situations, still done empirically or is based on experiments, yet some crashworthiness design capabilities have been developed and reported in the literature [7-9].

In general, design sensitivity analysis, DSA, since it involves the calculation of cost and performance functionals gradients w.r.t. the design variables represents a necessary step in the optimization process and provides an important tool for design improvement and "what if" studies.

The DSA for the dynamic response problem for structural and mechanical systems has been traditionally treated with separate formulations by Haug and Arora [10], Hsieh and Arora [11], Mene [12], Haug and Mani [13], Tortorelli an Lu [14], Choi and Wang [15]. More recently Cardoso and Arora [16] have proposed a unified approach for structural and mechanical systems using the adjoint structure concept. However the adjoint formulations, yield into a terminal value problem that needs to be integrated backwards in time resulting in a more complicated computational implementation as compared to the direct differentiation method which has been adopted herein.

In this paper a crashworthiness design methodology is presented where a structure is approximated by a multibody constrained mechanical system with revolute joints and non-linear revolute springs representing plastic hinges. This type of models have been successfully used in vehicle crash simulations [17-19]. For selected structural components, design variables are defined representing the corresponding linear elastic bending stiffness and the associated limit plastic moments.

2. Dynamic Analysis of Multibody Systems

For planar systems a 3 dimensional vector of Cartesian generalized coordinates, q^i, is defined for each rigid body: one rotational and two translational coordinates. The configuration of a mechanical system can be described by the *nc* vector of Cartesian coordinates, q, containing the generalized coordinates of all bodies in the system.

These coordinates are not all independent due to the existence of constraints representing different kinematic joints, such as spherical, revolute or translational joints. The corresponding *nh* constraint equations can be written in the form

$$\Phi \equiv \Phi(q,t) = [\Phi_1(q,t), \Phi_2(q,t), \ldots \Phi_{nh}(q,t)]^T = 0 \tag{1}$$

Differentiating (1) twice w.r.t. time

$$\dot{\Phi} \equiv \Phi_q \dot{q} + \Phi_t = 0 \tag{2}$$

and

$$\ddot{\Phi} \equiv \Phi_q \ddot{q} - \gamma \tag{3}$$

where $\gamma = -\left(\Phi_q \dot{q}\right)_q \dot{q} - 2\Phi_{qt}\dot{q} - \Phi_{tt}$.

The Lagrange equations of motion for the multibody system can be written as

$$M\,\ddot{q} + \Phi_q^T \lambda = g \tag{4}$$

where M is the *nc*nc* generalized mass matrix, g is the *nc* generalized force vector and λ is the nh vector of Lagrange multipliers. Equations (3) and (4) can be organized in the matrix form

$$\begin{bmatrix} M & \Phi_q^T \\ \Phi_q & O \end{bmatrix} \begin{Bmatrix} q \\ \lambda \end{Bmatrix} = \begin{Bmatrix} g \\ \gamma \end{Bmatrix} \tag{5}$$

forming a set of differential algebraic equations which has to be solved for the accelerations $\ddot{q}$. The vector of Lagrange multipliers λ is a by product of the solution that can be used with great advantage to obtain the reaction forces in the joints. The reader is referred to reference [20] for more details.

Using a proper integration scheme [21], equation (5) can be integrated in time to obtain the velocities $\dot{q}$ and the positions q.

3. Design Sensitivity Analysis

The design problem can be stated in following form

$$\text{Minimize } \Psi^o(b) \tag{6}$$

Subject to the constraints

$$\Psi_i^c(b) \Rightarrow \begin{cases} = 0 & i = 1,k \\ \le 0 & i = k+1,ncon \end{cases} \tag{7}$$

$$b^l \le b \le b^u \tag{8}$$

where $\Psi^o(b)$ and $\Psi^c(b)$ are functionals representing the objective function and the constraints respectively, and *ncon* is the number of constraints. b^l and b^u represent technological lower and upper bounds on the design variables and may also reflect acceptable structural designs for the elastic structure under normal loading conditions. $\Psi^o(b)$ and $\Psi_i^c(b)$ are functions of the *nb* design variables $b = \{b_1, b_2, \ldots b_{nb}\}^T$. b is also function of the state variables which include positions, velocities, accelerations and Lagrange multipliers, thus

$$\Psi = [\Psi^o, \Psi^c]^T = \Psi(q, \dot{q}, \ddot{q}, \lambda, t, b) \tag{9}$$

During the optimization process the cost and constraint functional must be evaluated as well as the corresponding gradients w.r.t. the design variables

$$\Psi_b^o = \frac{\partial \Psi^o}{\partial b} \quad \text{and} \quad \Psi_b^c = \frac{\partial \Psi^c}{\partial b} \tag{10}$$

A sensitivity matrix, L, can be defined such that

$$\delta\Psi = \begin{bmatrix} \frac{\partial \Psi^o}{\partial b} \\ \frac{\partial \Psi^c}{\partial b} \end{bmatrix} \delta b = L^T \delta b \tag{11}$$

Different types of constraints have to be considered which may correspond to different design criteria for the dynamic response problem [10]:

c1) Functionals of the particular type with the form

$$\Psi = \max\ f(q, \dot{q}, \ddot{q}, \lambda, t, b) \qquad \text{for } t^0 \le t^j \le t^1 \tag{12}$$

may represent, for example, the maximum acceleration which occurs during the impact in a particular point of interest in the structure only once and at distinct time t^j. t^0 and t^1 are the initial and final time of the impact period, respectively.

c2) Functionals of the integral type with the form

$$\Psi = \int_{t^0}^{t^1} f(q, \dot{q}, \ddot{q}, \lambda, t, b)\, dt \tag{13}$$

may represent either the total energy absorbed or a standard measure of vehicle crash severity index which is given by [7]

$$VCSI = \frac{1}{t^1 - t^0} \int_{t^0}^{t^1} a^2 dt \tag{14}$$

where -a(t) is the deceleration of the passenger compartment or the passenger itself when striking an obstacle, as for example, the front panel in a frontal collision or a door in a side impact situation.

Using the direct differentiation method, as proposed at reference [13], a first step for the design sensitivity analysis requires the linearization of the functionals (12) and (13). Without loss of generality just consider the first case

$$\delta\Psi = (f_q \delta q + f_{\dot{q}} \delta \dot{q} + f_{\ddot{q}} \delta \ddot{q} + f_\lambda \delta\lambda + f_b \delta b)\Big|_{t = t^k} \tag{15}$$

This equation requires the elimination of the dependency on the state variables. From a Taylor series expansion of the position vector

$$q(t, b+db) = q(t,b) + q_b\,\delta b + q_{bb}\delta b\,\delta b^T + .. \tag{16}$$

the first variation can be obtained as

$$\delta q = q_b\,\delta b \tag{17}$$

Similarly, the first variations for the other state variables can be obtained as

$$\delta \dot{q} = \dot{q}_b\,\delta b \tag{18}$$

$$\delta \ddot{q} = \ddot{q}_b\,\delta b \tag{19}$$

$$\delta \lambda = \lambda_b\,\delta b \tag{20}$$

For design sensitivity analysis the kinematic constraint equations (1) are also functions of the design variables, thus

$$\Phi = \Phi(q, t, b) = 0 \tag{21}$$

Calculating the first variation and taking into account (17)

$$\Phi_q q_b = -\Phi_b \tag{22}$$

Differentiating (22) twice w.r.t. time the sensitivity equations for the velocities and accelerations are obtained

$$\Phi_q \dot{q}_b = -(\Phi_q \dot{q})_q\, q_b - \dot{\Phi}_b \tag{23}$$

$$\Phi_q \ddot{q}_b = -\ddot{\Phi}_q q_b - 2\dot{\Phi}_q \dot{q}_b - \ddot{\Phi}_b \tag{24}$$

The generalized mass matrix and the generalized force vector are also functions of the design variables, thus

$$M = M(q, b) \tag{25}$$

and

$$g = g(q, \dot{q}, t, b) \tag{26}$$

Taking the first variation of the system dynamic equations of motion (5)

$$M\ddot{q}_b + \Phi_q^T\lambda_b = g_q q_b + g_{\dot{q}}\dot{q}_b + g_b - (M\ddot{q})_q q_b - (M\dot{q})_b - (\Phi_q^T\lambda)_q q_b + (\Phi_q^T\lambda)_b \tag{27}$$

Equations (24) and (27) can be written in matrix form as:

$$\begin{bmatrix} M & \Phi_q^T \\ \Phi_q & 0 \end{bmatrix} \begin{Bmatrix} q_b \\ \lambda_b \end{Bmatrix} = \begin{Bmatrix} g_q q_b + g_{\dot{q}}\dot{q}_b + g_b - (M\dot{q})_q q_b - (M\dot{q})_b - (\Phi_q^T\lambda)_q q_b + (\Phi_q^T\lambda)_b \\ -\dot{\Phi}_q q_b - 2\dot{\Phi}_q\dot{q}_b - \ddot{\Phi}_b \end{Bmatrix} \tag{28}$$

which can be integrated in time with the initial conditions

$$\begin{cases} q_b(t^0) = q_b^0 \\ \dot{q}_b(t^0) = \dot{q}_b^0 \end{cases} \tag{29}$$

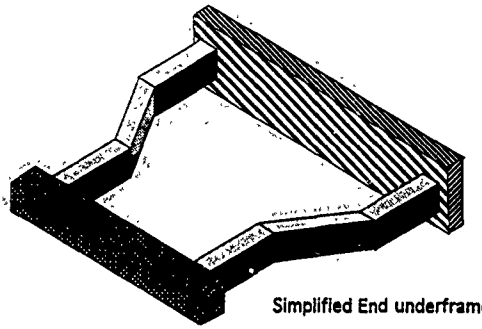
Since equations (5) and (28) have the same left hand side, they can be solved simultaneously to obtain the state variables q, $\dot{q}$, $\ddot{q}$, λ and the sensitivities q_b, $\dot{q}_b$, $\ddot{q}_b$, λ_b. This fact allow a greater efficiency in terms of computational efforts. The right hand side of the equation (28) has obtained using symbolic computation. Some examples of the equations obtained by symbolic computation, using the symbolic manipulation program MATHEMATICA [22], are shown in the Appendix.

3.1. OPTIMIZATION PROCEDURE

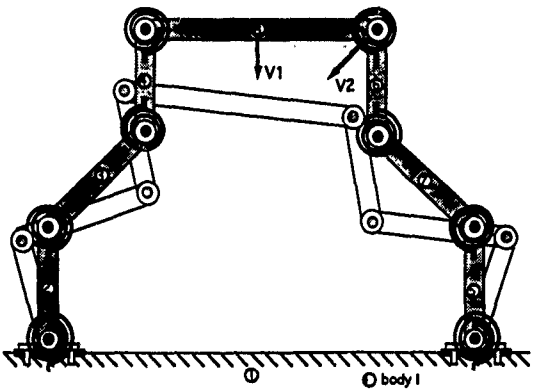
Several optimization algorithms, such as the feasible direction method, the gradient projection method, and the linearization method are coded in a multi-purpose program ADS (advanced Design Synthesis) [23], which was used in the present work. A modified projected gradients method [24] was selected in the present numerical examples.

4. Crashworthiness Optimization of an End Underframe Structure

To simulate the frontal structure of an end underframe of a railcar, a mechanical model was developed as shown in figure 1 where V1 and V2 represent initial velocities for a frontal and oblique collision, respectively. In the geometric discontinuities of the structure, eight plastic hinges have been modeled to simulate the plastic behavior of the structure. Each plastic hinge has a torsional spring mounted over a revolute joint connecting two consecutive bodies.



Simplified End underframe



Simplified Mechanical Model

Figure 1. Structural and mechanical model for an end underframe

The non linear response of the torsional springs is described in terms of a quasi static moment angle relationship based on the curves for plastic behavior proposed by Anderson et al.[25].

The design criteria for crashworthiness in the present case has been based only on the structural response predicted by the crashworthiness analysis for each design problem. Consequently, an objective function is defined as the maximum acceleration experienced by the frontal part of the structure. Other objective functions can be defined associated with the railcar crash severity index VCSI.

The bending stiffness K and the plastic moment M1 for the plastic hinges have been chosen as design variables. These two mechanical characteristics which are obtained for each cross section and are illustrated in figure 2 will give the necessary information for a proper choice of optimum design cross sections.

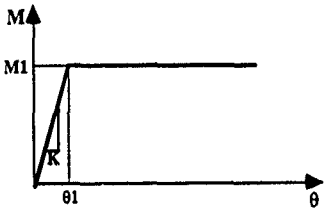


Figure 2. Torsional spring characteristics for the plastic hinges

Unilateral constraints are also defined to impose limits on the maximum structural crash deformation in order to preserve a minimum survival space in the car compartment.

Three crash situations have been simulated at a velocity of 6.88 m/s. The first one (S1) is a frontal impact, the second (S2) and the third (S3) are oblique collisions where the initial velocity V2 make an angle of 30° with the longitudinal direction. In crash situations S1 and S2 only longitudinal displacements constraints are imposed, and for S3 an additional lateral constraint also has been imposed.

The objective function and constraints, could be presented as

Find **b** such that $(\ddot{y}_g(t,\mathbf{b}))$ is minimized

Subject to: max. $(|\text{Displacement}(t,\mathbf{b})| - \bar{X}) \leq 0$

The maximum displacements X have respectively the values, X_{S1}=0.879, X_{S2}=0.479m, X_{S3}=0.279m and X_{S3}=0.3605m to the lateral constraint. For the impact situations S2 and S3 a simulation time t=0.1s has used.

Two design variables are considered as $\mathbf{b} = [b_1, b_2] = [\theta_1, M_1]$, where θ_1 and M_1 characterize the slope K and the plastic moment. The initial design vector is $\mathbf{b}$= [0.0873, 4644] and the lower and upper bounds are, $\mathbf{b}_l$ = [0.001, 800] and $\mathbf{b}_u$ = [0.1, 8000].

For the impact situation S1, the results of the displacement and acceleration for the initial and optimal solution, are shown in figures 3 and 4, respectively. This results show that the decrease in the maximum acceleration has achieved with a larger deformation.

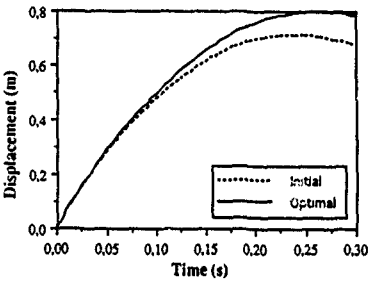


Figure 3. Displacement for the impact situation S1

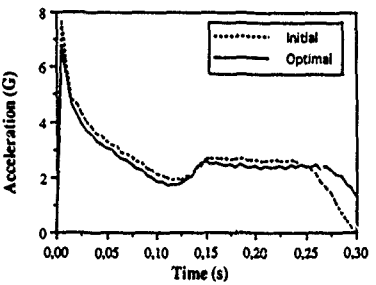


Figure 4 . Acceleration for the impact situation S1

The results of the displacement and acceleration for the impact situation S2, are shown in figures 5 and 6 respectively.

The substantial decrease in acceleration levels resulted from the fact that the specified displacement constraints yield an optimum solution corresponding to a very compliance structure.

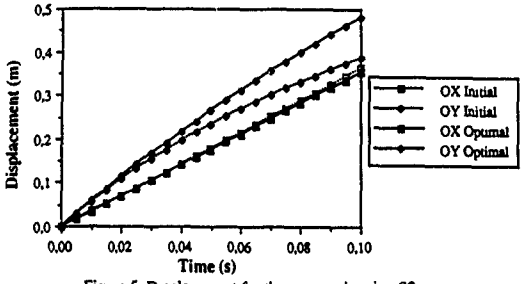


Figure 5. Displacement for the impact situation S2

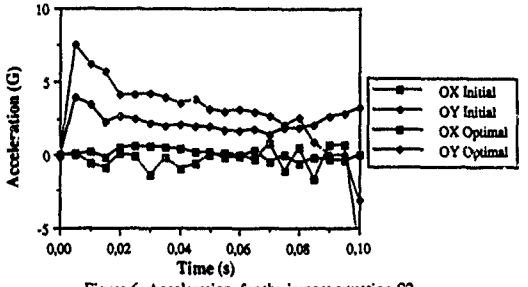


Figure 6. Acceleration for the impact situation S2

The results of the displacement and acceleration for the impact situation S3, are shown in figures 7 and 8 respectively.

In this case, a very strong longitudinal constraint were imposed, resulting in a constraint violation for the initial design. The optimal solution corresponds to a more stiff structure resulting in higher levels of acceleration.

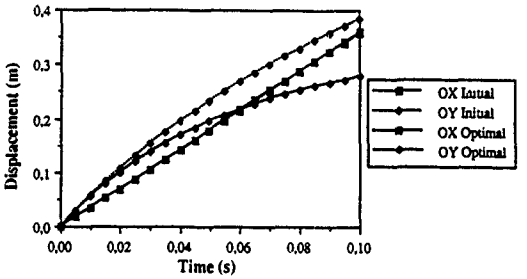


Figure 7. Displacement for the impact situation S3

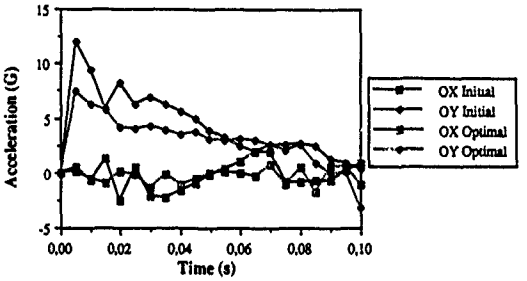


Figure 8. Acceleration for the impact situation S3

Table 1 summarizes the results obtained in cases S1, S2 and S3. The results clearly show that decrease in acceleration levels are obtained by relaxing the amount of structural permanent deformation during impact. On the other side in case S3, the lower deformation required in optimal design implies a higher level of acceleration.

In optimal design, all the constraints were satisfied, and the convergence of the optimization process has verified, for all the impact situations.

The results of the optimal design variables, allow to choose optimal cross sections to the end underframe structure.

Table 1 - Crashworthiness Problem of The End Underframe

	Crash Situation					
	S1		S2		S3	
	Initial	Optimal	Initial	Optimal	Initial	Optimal
Objective Function (G)	7.49	6.56	7.41	3.84	7.41	11.74
Cons.Dir.OY	-1.7×10^{-1}	-8.0×10^{-2}	-9.4×10^{-2}	-3.4×10^{-5}	1.1×10^{-1}	-1.3×10^{-3}
" OX	---	---	---	---	6.8×10^{-4}	-1.8×10^{-5}
b1 = θ1	0.0873	0.1	0.0873	0.0985	0.0873	0.0998
b2 = M1	4644.7	4327.0	4644.7	2521.6	4644.7	7408.2
CPU Time (s) (VAX 9000)	5206		4017		3626	

5. Conclusions

An efficient design methodology was developed for the purpose of studying an optimal vehicle body structure in a crash situation.

This methodology has been developed within the framework of mathematical programming methods involving an efficient dynamic analysis capability where a vehicle structure is modeled by rigid bodies connected by plastic hinges to modulate large structural deformations.

The capability of this methodology has been demonstrated for a end underframe structure in two impact situations, a frontal impact and a oblique impact.

Referer.ces

1. M. M. Kamal, "Analysis and Simulation of Vehicle to Barrier Impact", International Automobile Safety Conference, SAE Paper No. 700414, 1970.
2. A. B. Pifko and R. Winter, "Theory and Applications of Finite Element Analysis to Structural Crash Simulation", J. Computers and Structures, 13, pp277-285, 1981.
3. R. J. Hayduk, R. Winter, A. B. Pifko and E.L. Fesanella, "Application of the Non-linear Finite Element Computer Program DYCAST to Aircraft Crash Analysis", Structural Crashworthiness, Eds. N. Jones and T. Wierzbicky, Butterworths, London, England, pp.283-307, 1983.
4. E. Haug, "The PAMCRASH Code as an Efficient tool for Crashworthiness Simulation and Design", Second European Cars/trucks Simulation Symposium, Schliersee, Germany, May 22-24, 1989.
5. J. O. Halquist, "Theoretical Manual for DYNA-3D", Lawrence Livermore Laboratory, 1982.
6. T. Belytschko and J. M. Kenedy, "WHAMS-3D, An Explicit 3D Finite Element Program", KBS2 Inc. P.O. Box 453, Willow Springs, IL 60480, 1986.

7. J. A. Bennett, K. H. Lin and M. F. Nelson, "The Application of Optimization Techniques to Problems of Automobile Crashworthiness", SAE Trans. 86, pp2255-2262, 1976.
8. J. O. Song, "An Optimization Method for Crashworthiness Design", Proc. Sixth Int. Conf. Vehicle Structural Mechanics, Detroit, MI, pp39-46, 1986.
9. R. Lust, "Structural Optimization with Crashworthiness Constraints", Structural Optimization, 4, pp85-89, 1992.
10. E. J. Haug and J. S. Arora, "Applied Optimal Design", N. Y., Willey, 1979.
11. C. C. Hsieh and J. S. Arora, "Design Sensitivity Analysis and Optimization of Dynamic Response", Comp. Meth. Appl. Mech. Engrg., 43, pp195-219, 1984.
12. R. A. Meric, "Shape Design Sensitivity Analysis of Dynamic Structures", AIAA J., 26, pp206-212, 1988.
13. E. J. Haug, N. K. Mani and P. Krishnaswami, "Design Sensitivity Analysis and Optimization of Dynamically Driven Systems", Proc. NATO ASI Computer-Aided Analysis and Optimization of Mechanical Systems Dynamics, 1984.
14. D. A. Tortorelli and S. C. Lu, "Design Sensitivity Analysis for Elastodynamic Systems", Mech. Struct. & Mach, 18, pp77-106, 1990.
15. K. K. Choi and S. Wang, "Continuum Design Sensitivity Analysis of Structural Dynamic Response using Ritz Sequence", Proc. AIAA/ASME/ASCE/AMS/ACS 31st Structures, Structural Dynamics and Materials Conf. Part 1, Long Beach, CA, pp385-393, 1990.
16. J. B. Cardoso and J. S. Arora, "Design Sensitivity Analysis of Nonlinear Dynamic Response of Structural and Mechanical Systems", Structural Optimization, 4, pp37-46, 1992.
17. P. Nikravesh, I. Chung and R. L. Benedict, "Plastic Hinge Approach to Vehicle Crash Simulation", Comp. & Struct., 16, No.1-4, pp395-400, 1983.
18. J. Ambrósio, P. Nikravesh and M. S. Pereira, "Crashworthiness Analysis of a Truck", J. Mathematical Computer Modelling, 14, pp959-964, 1990.
19. M. S. Pereira, P. Nikravesh, G. Gim and J.Ambrósio, "Dynamic Analysis of Roll-over and Impact of Vehicles", XVIII Bus and Coach Experts Meeting, Budapest, Hungary, 1987.
20. P. E. Nikravesh, "Computer Aided Analysis of Mechanical Systems", Prentice-Hall, 1988.
21. L. F. Shampine, and M. K. Gordon, "Computer Solutions of Differential Ordinary Equations", Prentice Hall, New Jersey, 1975.
22. S. Wolfram, "Mathematica: A System for Doing Mathematica By Computer", Addison-Wesley, 1988.
23. G. Vanderplaats, "ADS- A Fortran Program for Automated Design Synthesis", Version 2.01, Eng. Design Optimization Inc., Ca, USA, 1987.
24. G. Vanderplaats, and F. Moses, "Structural Optimization by Methods of Feasible Directions", J. Comp. & Struct., 3, pp739-755, 1973.
25. W. J., Anderson, I. K. McIvor and B. S. Kimball, "Modular Program Development for Vehicle Crash Simulation", Vol. 2; Plastic Hinge Experiments", DOT-HS-802-531, 1977.

Appendix

In this appendix the general form of some terms of the equations of sensitivities (28) is shown. These terms have been obtained using the symbolic manipulation program MATHEMATICA.

The first column is for a linear spring, as an example of an actuator, and the second for a translation joint as an example of a kinematic joint.

g_b

```
GB(1,1)=ELX*DELEL/EL
GB(1,2)=ELY*DELEL/EL
GB(1,3)=XPIMXI*ELY*DELEL/EL
    -YPIMYI*ELX*DELEL/EL
GB(1,6)=-(ELY*XPJMXJ* DELEL/EL)
    +YPJMYJ*ELX*DELEL/EL
GB(2,1)=ELX*ELD
GB(2,2)=ELY*ELD
GB(2,3)=XPIMXI*ELY*ELD/EL -
    YPIMYI*ELX*ELD/EL
GB(2,6)=-(ELY*XPJMXJ*ELD/EL)
    +YPJMYJ*ELX*ELD/EL
```

g_q

```
GQ(1,1)=ELX*(-(ELX*SP(K,5)/EL) -
    ELXD*SP(K,6)/EL+ ELX* -
    (ELX*ELXD +ELY*ELYD)*SP(K,6)/
    (EL**2)**3/2)/EL
    +ELX**2*FF/(EL**2)**3/2 - FF/EL
GQ(1,2)=ELX*(-(ELY*SP(K,5)/EL) -
    ELYD*SP(K,6)/EL+ ELY* -
    (ELX*ELXD +ELY*ELYD)*SP(K,6)/-
    (EL**2)**3/2)/
    EL+ELY*ELX*FF/(EL**2)**3/2
GQ(1,3)=ELX*((2*-XPIMXI*ELX)*SP(K,5)/
    (2*EL)- (-(QD(3,L)*-YPIMYI*ELY)+
    QD(3,L)*XPIMXI*- ELX+YPIMYI*
    ELXD-XPIMXI*ELYD)*SP(K,6)/- EL-
    (2*-XPIMXI*ELX)*(ELX*ELXD
    +ELY*ELYD)*- SP(K,6)/(2*(EL**2)
    **3/2))/ EL - ELX*- (2*-XPIMXI*ELY +
    2*YPIMYI*ELX)*FF/- (2*(EL**2)**3/2)
    + YPIMYI*FF/EL
...
```

$g_{\dot{q}}$

```
GQD(1,1)=-(ELX**2*SP(K,6)/EL**2)
GQD(1,2)=-(ELY*ELX*SP(K,6)/EL**2)
GQD(1,3)=ELX*(-(XPIMXI*ELY)
    +YPIMYI*ELX)*SP(K,6)/EL**2
...
```

$(\Phi_q \dot{q})_q$

```
FQQDQ[1,1]=QD(3,L)*UI
FQQDQ[1,2]=QD(3,L)*VI
FQQDQ[1,3]=-(QD(1,J)*UI) + QD(1,L)*UI -
    QD(2,J)*VI + QD(2,L)*VI - QD(3,J)*
    (-VI*XPJMXJ + UI*YPJMYJ) - QD(3,L)*
    (-YPIMYI*UI + 2*XPIMXI*VI -
    XPIMXI*VI + VI* - (Q(1,J) - Q(1,L)
    -XPIMXI+ XPJMXJ) -UI* - (Q(2,J) -
    Q(2,L) -YPIMYI+ YPJMYJ))
...
```

$(\Phi_q \ddot{q})_q$

```
FQQDDQ[1,1]=QDD(3,L)*UI
FQQDDQ[1,2]=QDD(3,L)*VI
FQQDDQ[1,3]=-(QDD(1,J)*UI)+ QDD(1,L)*UI-
    QDD(2,J)*VI+ QDD(2,L)*VI- QDD(3,J)*
    (-VI*XPJMXJ + UI*YPJMYJ)-QDD(3,L)
    *(-(YPIMYI*UI)+ 2*XPIMXI*VI-
    XPIMXI* VI- VI*(Q(1,J)- Q(1,L)-
    XPIMXI+ XPJMXJ) +UI*(Q(2,J)- Q(2,L)-
    YPIMYI+ YPJMYJ))
...
```

$[(\Phi_q \dot{q})_q \dot{q}]_q$

```
FQQDQQDQ[1,1]=-QD(3,L)**2*VI
FQQDQQDQ[1,2]=QD(3,L)**2*UI
FQQDQQDQ[1,3]=-QD(2,J)*QD(3,L)*UI+
    QD(2,L)*QD(3,L)*UI-
    QD(1,J)*QD(3,L)*VI-
    QD(1,L)*QD(3,L)*VI- QD(3,J)*(QD(3,L)*
    (-UI*XPJMXJ- VI*YPJMYJ)-
    QD(3,J)*(UI*XPJMXJ+ VI*YPJMYJ))+
    QD(3,L)*(QD(2,J)*UI+
    QD(2,L)*UI+QD(1,J)*VI-QD(1,L)*-VI-
    QD(3,J)*(-UI*XPJMXJ - VI*YPJMYJ)-
    QD(3,L)*(-2*XPIMXI*UI+3*XPIMXI*UI
    +YPIMYI* VI- UI* (Q(1,J)- Q(1,L)-
    XPIMXI+ XPJMXJ)-VI*(Q(2,J)- Q(2,L)-
    YPIMYI+ YPJMYJ)))
...
```

THE MULTIDISCIPLINE DESIGN CONCEPT FOR MECHANISMS

O.I. Sivertsen
The University of Trondheim
The Nowegian Institute of Technology
N - 7034 Trondheim
Norway

T. Rølvåg
SINTEF Production Engineering
N - 7034 Trondheim
Norway

H.P. Hildre
The University of Trondheim
The Norwegian Institute of Technology
N - 7034 Trondheim
Norway

ABSTRACT. This paper describes a design concept for multi body systems developed in the ESPRIT II #5524 project called "High performance computing for multidiscipline dynamic simulation of mechanisms" (MDS)[1]. The concept is based on a simulation tool combining multi body simulation, the finite element method and control engineering (FEDEM)[2]. The tool is developed further within the MDS project to complete an automatic design optimization loop including dynamic simulation and sensitivity analysis. The project is aiming for an interactive design environment based on simultaneous

1 Running from Nov. 1990 to April 1993 with partners from Germany, Italy, England, Sweden, Norway and coordinated by the Danish Technological Institute in Denmark.

2 FEDEM - Finite Element Dynamics of Elastic Mechanisms, A multi-discipline simulation tool for mechanisms developed by The Norwegian Institute of Technology and SINTEF Production Engineering, Norway.

simulation and result presentation as curve plotting and animation. The designer may also inspect the results as animations of mechanism motion including scaled deformations of bodies and stress contour fringes. A Motif-based program manager tool is developed to control program execution and data flow between the different modules involved, that simplifies the designers interaction with the system. Also a STEP toolkit is developed to standardize data communication between internal and external program modules. There has been developed an interface for simultaneous dynamic simulation between the internal multidiscipline simulation tool and an external simulation tool like for instance MATLAB[3], with the simulation model distributed between the programs.

1. INTRODUCTION

The MDS consortium were based on the software packages:

- FEDEM - for multidiscipline dynamic simulation of mechanisms based on a nonlinear finite element formulation.
- FEMGEN / FEMVIEW - for finite element pre-and post-processing
- KISMET - for high performance animation
- OPOS - for Optimization analyses
- SESAM - the FEM analysis system

Based on the software packages mentioned above the objectives for the project were:

- Combining and extending the actual software packages for multidiscipline Simulation, FEM pre-and post-processing, mechanism animation and optimization analyses.
- Extending the functionality of the simulation software according to the application partner's needs.
- Extending the modelling and visualization capabilities within the field of mechanisms, including an overall Motif-based user interface.
- Developing STEP interfaces for data communication

[3] MATLAB - A commercial available simulation system.

- Developing an optimization tool for mechanisms.
- Applying the software within the fields of robotics, aerospace, SME- applications etc.

2. THE MULTIDISCIPLINE SIMULATION SOFTWARE

The simulation system FEDEM is a general purpose simulation system for multidiscipline dynamic simulation of mechanism motion combining a non-linear finite element formulation with mechanism and control analysis. The different links of a mechanism are modelled flexible from a library of beam, shell and solid finite elements. The links are joined together by a set of joint types. Liner and non-linear springs and dampers may be included in a model between links and in joints. Time dependent motion and loading may be included on links and in joints. From a library of control elements, a control system for the mechanism may be modelled, taking the measurements from the mechanism variables and generating actuator forces on the mechanism.

Based on the user specified requirements, extended modelling functionality were developed [3,4,6,7,12,13,21]. The new modelling functionality developed within the project are, new and improved joint and transmission models, joint and gear friction, user subroutine interface, extended control element library etc. A typical FEDEM feature is the flexible sliding joints based on the prismatic and cylindric joint primitives. These may be used to model for example flexible telescopic joints.

A simultaneous interface to an external controller simulation package are also developed and implemented with MATLAB as the external simulation package. The interface is based on a general purpose socket communication for a computer network, and the simulation codes (in this case FEDEM and MATLAB) may run on the same or on different computers in the network. For the external MATLAB package a user written subroutine is added. In this way the control engineer may model the control system in a simulation package that he is familiar with as an alternative to modelling also the control model directly in FEDEM. However, longer computer times will be the penalty.

A functionality for generating modal animation of mechanism deflection is developed, and also rigid body kinematic simulation is made available in FEDEM.

3. MODELLING AND RESULT VISUALIZATION

For the multidiscipline dynamic simulation, the links are modelled as FEM substructures. Within the project the general purpose FEM preprocessor FEMGEN are used for generating the FEM models for the different links of a multi body system. Conversion

of the FEM models to the input format of the simulation program is straight forward as for any general purpose FEM program.

Modelling the system model for a mechanism is usually not a functionality of a FEM preprocessor, and a new module is developed within the project for this purpose. This mechanism modeler may import the FEM models for the different bodies of the mechanism from the FEM preprocessor, in this case FEMGEN. It is also of interest to export coordinates of points on each link relative the link coordinate system, for instance for joints, springs, dampers, loads etc., to the FEM preprocessor as external points for the FEM mesh. The mechanism modeler will display simplified drawings for the different bodies and present joints, springs, dampers, loads etc., as graphic symbols on the display. Control models integrated in a mechanism are generated through numerical input, however, these data will also be entered graphically in the next version of the mechanism modeler.

A large amount of different kind of results are generated during a dynamic simulation run. The result are generated on two levels, the mechanism and the link level, as for the modelling. Typical result on the mechanism level are positions, velocities, accelerations, forces, torques in external nodes, joints, springs and dampers. These type of data are most conveniently presented as variable-time, or variable-variable curve plots. Also variation of eigenvalues at system level and control variables are most conveniently presented as curve plots. However, animation of motion of the simulated multi body system is a very important tool for the designer (analyst) as a first check of the simulated results, especially when the simulation and visualization models are the same. Carrying out some extra FEM processing of the results, the animation of motion may also include scaled deformations of bodies and/or Von Mises stresses on the body surfaces. This gives the designer very good insight into the dynamics of his system.

If the simulation also has specified eigenvalue analysis for certain time incidents, the eigenvectors may be used to visualize eigenmodes of the system. The first, second, third etc. eigenmodes for a certain position may be visualized as mode animation by scaling the eigenvector by a sinus function.

The FEM results for the different parts of a mechanism may be displayed individually in a FEM postprocessing program after converting the results to the actual visualization programs database, as within this project is FEMVIEW. All facilities for FEM visualization within the FEM postprocessor is then available for result visualization. Also FEM results on the assembled mechanism may be visualized in the FEM postprocessor.

Because of the large models and amount of results in a FEM analysis, traditionally the analysis is divided in preprocessing, FEM analysis and postprocessing. However, for multidiscipline simulation in the MDS project, we are investigating the possibilities for simultaneous analysis and result visualization on a high performance computer system using process to process communication between the simulation program and a set of

visualization programs for animation of motion, curve plotting etc [1,5,10,18,23,24]. The user is in this way interactively controlling simulation parameters during simulation, and if the simulation model is showing unexpected behavior, the simulation may be stopped and checked more in detail before continued, possibly with modified integration parameters. Alternatively, the simulation is terminated and the simulation model is modified before a new simulation is initiated. In this way the user is in full control of his simulation with better understanding of a simulation model and fewer failure simulations. The less productive waiting times during off line simulations will in this way be eliminated, and the user may concentrate on the most resent results as they are generated.

Within the MDS project this philosophy is only tested for relatively small simulation models because this approach is very much dependent on the computer power available for simulation and visualization. To make this a working solution for larger models, much more computing power must be available for instance through parallelization of the simulation and visualization codes between a number of high performance CPU's. The super element approach for the simulation model makes it very well suited for parallelization, especially if also stress analysis are to be executed and visualized simultaneously. Certainly, this working philosophy is feasible within the evolving computer technology for relatively large simulation models, and will be a research area of high priority.

For multidiscipline simulation, a wide variety of computer programs are involved for model generation, simulation and result visualization, and to make a unified working environment for the user, the MDS-Program Manager (MDS-PM) is controlling the different programs that are made available to the user and are keeping track of the corresponding data and the transformation of data between the programs [8,16,17]. Program modules supplied by the consortium, here called internal programs, are fully controlled by the MDS-PM. External programs, that are programs not controlled by the MDS-consortium, are controlled by generating a window on the workstation for starting the program. These programs are then living quite independently of the MDS-PM until their termination are reported back, and the workstation window is deleted. The MDS-PM will set up dependencies and sequences between the programs and the corresponding data models used, and when some data are changed within some computation sequence, color codes are used to inform the user of which programs has to be re-executed to complete the simulation with this updated data.

4. DATA COMMUNICATION THROUGH STEP ETC.

A large number of data interfaces for the multidiscipline simulation are identified within the MDS-project [2] as FEM models, mechanism models, control models, intermediate simulation models, intermediate simulation results, FEM results, surface models and

transformation matrices for animation of motion, time series results for mechanism and control variables etc. All these data should be handled in a unified way and controlled by the MDS-PM. For the purpose of handling and controlling all these data the MDS-STEP toolkit [15] were developed. The toolkit is intended to aid an applications programmer in the accessing of STEP data. Some of the tools provided are useful in the development of EXPRESS data models and validation and printing of STEP exchange files. The toolkit comprises stand alone modules and a library of functions to access STEP data.

For most of the data interfaces identified above in the MDS-project, EXPRESS data models are developed and implemented through the STEP toolkit. For many of the data models identified no standardized application protocols are available as for mechanism and control data, and working STEP models for the MDS-project are then developed. However, where standards are evolving, as for FEM data, the STEP models used within the project comply with these models. See Figure 1.

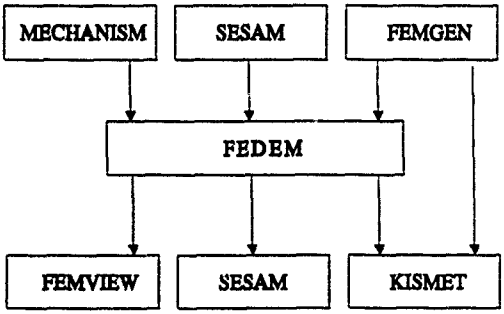


Figure 1 *MDS STEP model overview*

A new multidiscipline input format for modelling at system level, are developed. The aims of this new format are to be more object oriented, and to have as few redundancies as possible. This format makes mechanism modelling a quite straight forward task for the user. Table 1 shows the entity names used in this format and Table 2 shows an example four-bar mechanism model.

Table 1 *Mechanism entities*

NAME	# ENTITIES	DESCRIPTION
MECHANISM	1	Model name, gravity constants,etc.
ANALYSIS	1	Program Control parameters, etc.
LINK	1 : #	Link position/orientation etc.
TRIAD	1 : #	Positions for coordinate systems used in joints, springs, dampers, loads etc.
JOINT	0 : #	Joint definitions
SPRING	0 : #	Axial and joint springs.
DAMPER	0 : #	Axial and joint dampers
MOTION	0 : #	Motion input
HIGHER_PAIR	0 : #	Modelling of transmissions
FUNCTION	0 : #	Functions defining input motions, forces, spring stiffness's etc.
CONTROL_MOD	0 : #	Input data for control module.
CONTROL_IO	0 : #	Definition of control inputs/outputs
SENSITIVITY	0 : 1	Definition of sensitivity output for optimization
ENDDATA	1	Indicates end of model.

Table 2 *Four-bar mechanism model example*

```
MECHANISM
        ID                      =  1
        MECH_DESCR      = 'Example 1 in FEDEM Example Manual'
        X_GRAV                  = 0.0000E+00
        Y_GRAV                  = -.9810E+01
        Z_GRAV                  = 0.0000E+00
        ;

ANALYSIS
        ID              = 1
        ANA_DESCR       = 'Path following mechanism'
        START_TIME      = 0.0000E+00
        END_TIME        = 0.1000E+01
        TIME_STEP       = 0.1000E-01
        TIME_INCR       = 0.1000E-01
        ;

LINK    ID              =  1
        LINK_DESCR      = 'Input Link'
        LCS_ORIGIN      = 0.0000E+00  0.0000E+00  0.0000E+00
        LCS_XPOINT      = 0.0000E+00  0.1000E+01  0.0000E+00
        LCS_YPOINT      = -.1000E+01  0.0000E+00  0.0000E+00
        FLEX_DAT_REF    =  1
        ;
TRIAD   ID              =  1
        TRIAD_DESCR     = 'Ground Triad'
        OWNER_LINK      =  1
        GRT_ORIGIN      = 0.0000E+00  0.0000E+00  0.0000E+00
        GRT_XPOINT      = 0.1000E+01  0.0000E+00  0.0000E+00
        GRT_YPOINT      = 0.0000E+00  0.1000E+01  0.0000E+00
        ;
TRIAD   ID              =  2
        TRIAD_DESCR     = 'Master Triad for Joint ID=2'
        OWNER_LINK      =  1
        GRT_ORIGIN      = 0.0000E+00  0.1500E+00  0.0000E+00
        GRT_XPOINT      = 0.1000E+01  0.1500E+00  0.0000E+00
        GRT_YPOINT      = 0.0000E+00  0.1000E+01  0 0000E+00
        ;

LINK    ID              =  2
        LINK_DESCR      = 'Coupler Link'
        LCS_ORIGIN      = 0.0000E+00  0.1500E+00  0 0000E+00
        LCS_XPOINT      = 0.4000E+00  0.6000E+00  0.0000E+00
        LCS_YPOINT      = 0.0000E+00  0.2000E+0.  0.0000E+00
        FLEX_DAT_REF    =  2
        LINK_MASS       = 0.4000E+00
        ;

TRIAD   ID              =  3
        TRIAD_DESCR     = 'Slave Triad for Joint ID=2'
        OWNER_LINK      =  2
        GRT_ORIGIN      = 0.0000E+00  0.1500E+00  0.0000E+00
        GRT_XPOINT      = 0.1000E+01  0.1500E+00  0 0000E+00
        GRT_YPOINT      = 0.0000E+00  0.1000E+01  0.0000E+00
        ;
TRIAD   ID              =  4
        TRIAD_DESCR     = 'Master Triad for Joint ID=3'
        OWNER_LINK      =  2
        GRT_ORIGIN      = 0.1000E+00  0.3750E+00  0 0000E+00
        GRT_XPOINT      = 0.1000E+01  0.3750E+00  0.0000E+00
        GRT_YPOINT      = 0.1000E+00  0.1000E+01  0.0000E+00
        ;
```

```
!       DESCRIPTION OF THE OUTPUT LINK

LINK    ID              =  3
        LINK_DESCR      = 'Output Link'
        LCS_ORIGIN      = 0.3000E+00  0.3750E+00  0.0000E+00
        LCS_XPOINT      = 0.3000E+00  0.0000E+00  0.0000E+00
        LCS_YPOINT      = 0.1000E+01  0.0000E+00  0.0000E+00
        FLEX_DAT_REF    =  3
        /

TRIAD   ID              =  5
        TRIAD_DESCR     = 'Slave Triad for Joint ID=3'
        OWNER_LINK      =  3
        GRT_ORIGIN      = 0.3000E+00  0.3750E+00  0.0000E+00
        GRT_XPOINT      = 0.1000E+01  0.3750E+00  0.0000E+00
        GRT_YPOINT      = 0.3000E+00  0.1000E+01  0.0000E+00
        /
TRIAD   ID              =  6
        TRIAD_DESCR     = 'Ground Triad'
        OWNER_LINK      =  3
        GRT_ORIGIN      = 0.3000E+00  0.0000E+00  0.0000E+00
        GRT_XPOINT      = 0.1000E+01  0.0000E+00  0.0000E+00
        GRT_YPOINT      = 0.3000E+00  0.1000E+01  0.0000E+00
        /

!       DESCRIPTION OF THE REVOLUTE JOINTS

JOINT   ID              =  1
        JOINT_DESCR     = 'Revolute joint to ground'
        JNT_TYPE        =  REV
        SLAVE_TRIAD     =  1
        /
JOINT   ID              =  2
        JOINT_DESCR     = 'General revolute joint'
        JNT_TYPE        =  REV
        SLAVE_TRIAD     =  3
        MASTER_TRIADS   =  2
        /
JOINT   ID              =  3
        JOINT_DESCR     = 'General revolute joint'
        JNT_TYPE        =  REV
        SLAVE_TRIAD     =  5
        MASTER_TRIADS   =  4
        /
JOINT   ID              =  4
        JOINT_DESCR     = 'Revolute joint to ground'
        JNT_TYPE        =  REV
        SLAVE_TRIAD     =  6
        /

!       DESCRIPTION OF THE SPRING WHICH IS GENERATING
!       THE INPUT MOTION. THE LENGTH OF THE SPRING IS
!       (INIT_LNTH + SCALE_LNTH * TIME) WHERE TIME IS
!       THE CURRENT SIMULATION TIME.

SPRING  ID              =  1
        SPR_DESCR       = 'Actuator spring'
        SPR_TYPE        = JOINT
        REF_JNT         =  1
        JOINT_DOF       = Z_ROT
        INIT_LNTH       = 0.0000E+00
        SCALE_LNTH      = -.6280E+01
        INIT_STIF       = 0.1000E+04
        SCALE_STIF      = 0.0000E+00
        /
ENDDATA
```

An EXPRESS data model for this format is about to be developed.

5. MECHANISM OPTIMIZATION

Optimization analysis are about to be a standard feature for the designer using conventional FEM programs. In mechanism design, especially flexible mechanisms, however, optimization analysis is a new feature that has up till now not been available. In the MDS project there has been done quite important developments towards an automatic or semiautomatic optimization loop for the design of flexible mechanisms [9,14,22].

For the search towards an optimum, sensitivity gradients from the simulation run is essential. A sensitivity is defined as the derivative of a response variable with respect to a so called "design parameter" in the mechanism model. A design parameter is a parameter in the model effecting one or more input data for the simulation. Typical input data that could be effected are stiffness', damping coefficients, additional masses, initial conditions, loading, parameters in functions etc.

Sensitivity requests defines output variables from a simulation where sensitivity calculation is required with respect to the design parameters. Typical sensitivity requests may be for positions, velocities, accelerations, spring or damper forces joint variables etc.

The sensitivities are calculated from perturbating the design variables one at a time in the input data and the sensitivities for all positions, velocities and accelerations of primary simulation variables are evaluated. These sensitivities are then used as the basis for calculating the requested sensitivities. The sensitivity analysis during dynamic simulation is quite efficient because the triangularized system matrices from the integration could be reused for the sensitivity analysis requiring only forward and backward substitution to solve for the sensitivities.

Requested responses and sensitivities may be calculated repeatedly in the optimization loop and the design variables are updated accordingly in the simulating model. The optimization algorithm is working with respect to a set of constraints for the simulation responses and some criteria defines when the optimization should be terminated. The results from the MDS project in this field is just the beginning of what seems to be a very promising and exciting tool for the designer in the future.

6. MULTIDISCIPLINE APPLICATIONS

Within the MDS project the simulation tools has been applied on the IRB6000 industrial robot [19,26] by ABB Robotics Products AB in Sweden, see Figure 2.

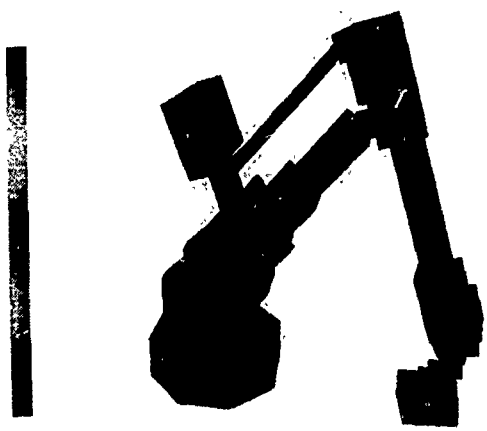


Figure 2 *The visualization model from the IRB6000 industrial robot simulation including stress contours*

Deployment of satellite booms, including latching [25], were simulated by Dornier GmbH in Germany, see Figure 3.

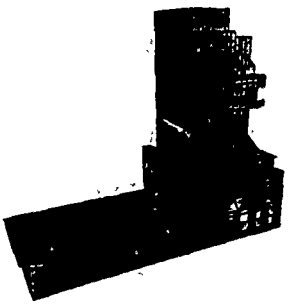


Figure 3 *The visualization model from the satellite boom deployment simulation.*

Machining center operations [31] are simulated by Syntax Factory Automation in Italy, see Figure 4.

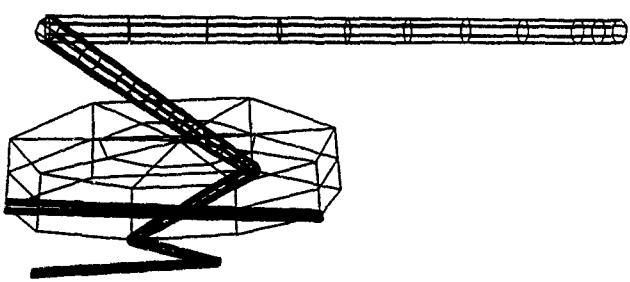


Figure 4 *The visualization model from the machining center simulation.*

7. CONCLUSIONS

The very ambitious objectives set up for the MDS project has to a large extent been fulfilled. New developments and applications has to some extent been running in parallel, and some of the new features developed towards the end of the project has not been tested fully in real applications.

The simultaneous simulation and visualization and the optimization and sensitivity analysis were planed as test implementations to investigate the potentials in these areas, and from the experience gained so far it is very promising.

For all the other objectives for the MDS developments, the resulting programs are in a state close to a commercialization standard.

8. REFERENCES

[1] Architecture and computer environment document, MDS deliverable D1101 (restricted).

[2] Standard interface format specification, MDS deliverable D1201 (public).

[3] Benchmark example and functional requirements, MDS deliverable D1301 (public).

[4] SACODY (ESPRIT #1561) and MDS common workshop, MDS deliverable D1302 (restricted).

[5] A user interface scenario, MDS deliverable D3101 (public).

[6] Multidiscipline software specification document 1, MDS deliverable D2101 (restricted).

[7] Updated multidiscipline simulation: User's guide, MDS deliverable D2103 (restricted).

[8] Specification of the user environment, MDS deliverable D3102 (public).

[9] Sensitivity theory and software requirement document, MDS deliverable D2201 (restricted).

[10] Technical specification for simultaneous result animation, MDS deliverable D3301 (restricted).

[11] Data sheets (CIM conference week Nov. 1992), MDS deliverable D4514 (public).

[12] Multidiscipline software specification document 2, MDS deliverable D2104 (restricted).

[13] Updated multidiscipline simulation: User's guide, MDS deliverable D2106 (restricted).

[14] Optimization theory and software requirement document, MDS deliverable D2203 (restricted).

[15] STEP toolkit documentation, MDS deliverable D2301 (public).

[16] Program manager; User's guide, MDS deliverable D3201.1 (restricted)

[17] Program manager; Resource file manual, MDS deliverable D3201.2 (restricted).

[18] FEM processors software documentation, MDS deliverable D3202 (due may 93, public).

[19] Verification by testing, MDS deliverable D4203 (restricted).

[20] SME-implementations, MDS deliverable D4501 (restricted).

[21] Software documentation document, MDS deliverable D2107 (due may 93, restricted).

[22] Software description document, MDS deliverable D2205 (due may 93, public).

[23] Final report, MDS deliverable D3302 (due may 93, public).

[24] Simultaneous simulation implementation in high performance computer environment, MDS deliverable D3401 (due may 93, restricted).

[25] Report of relevant results and experiences gained during this task, MDS deliverable D4101 (due may 93, public).

[26] Report of experience in robotics, MDS deliverable D4204 (due may 93, restricted/public).

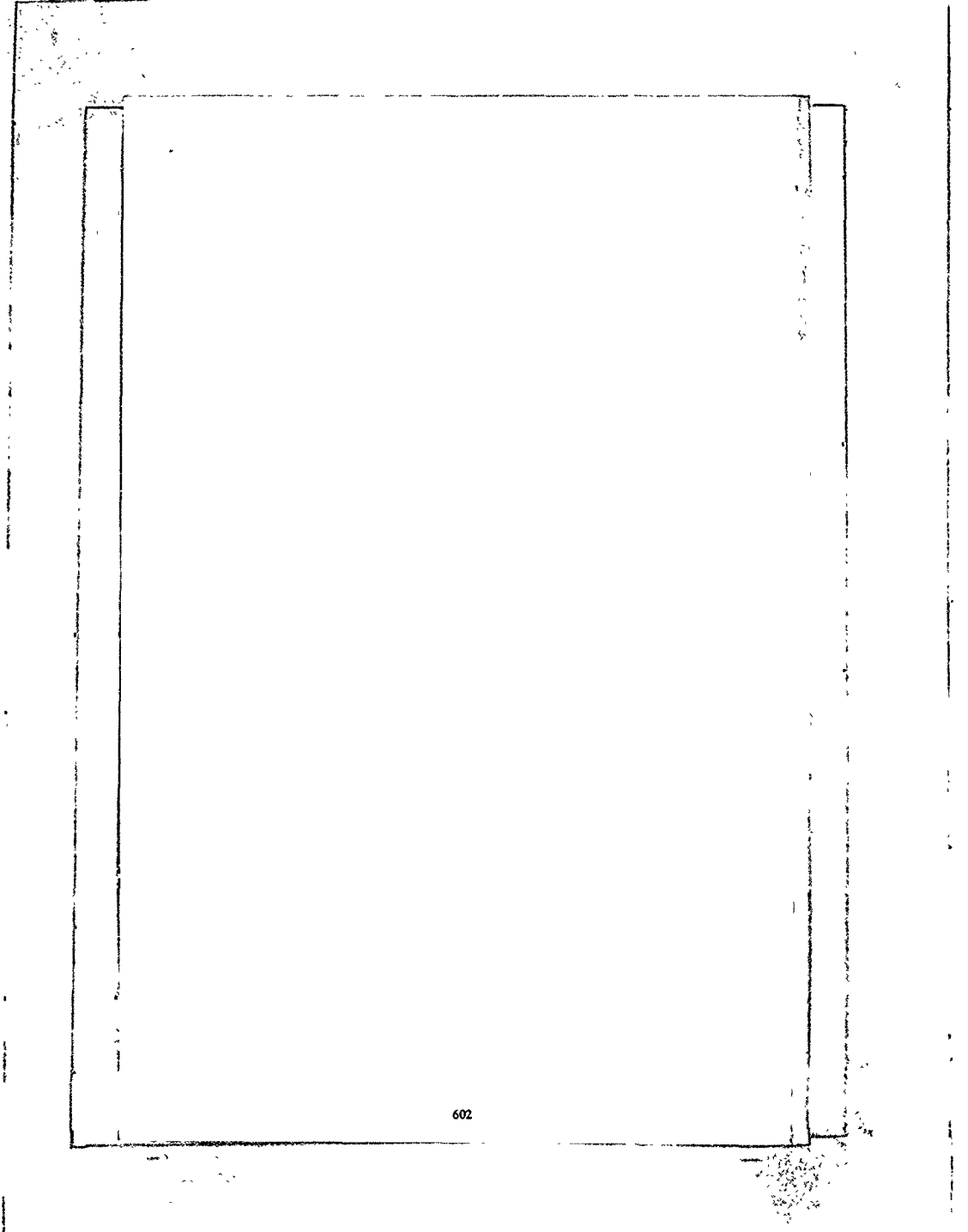
[27] Experience gained in aerospace applications, MDS deliverable D4301 (due may 93, public).

[28] Final report, MDS deliverable D4401 (due may 93, restricted).

[29] Experience in implementation SME, MDS deliverable D4502 (due may 93, public).

[30] Simulation guideline document, MDS deliverable D4503 (due may 93, public).

[31] Final report, MDS deliverable D4601 (due may 93, public).



MAIN SPONSOR

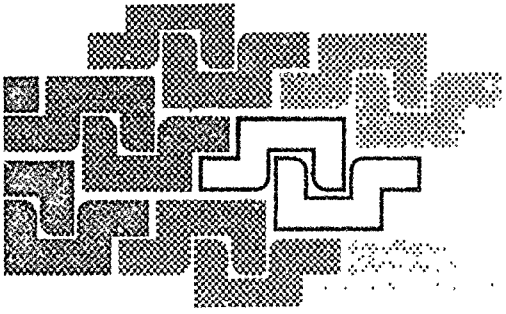
NATO - North Atlantic Treaty Organization

OTHER SPONSORS

U.S. Army TARDEC - United States Army, TARDEC
NSF - National Science Foundation
JNICT - Junta Nacional de Investigação Científica e Tecnológica
FLAD - Fundação Luso Americana para o Desenvolvimento

OTHER SUPPORTS

HEWLLET PACKARD PORTUGAL
RANK XEROX PORTUGAL
SOGAPAL - Sociedade Gráfica da Paiã
LUSANOVA - Agência de Viagens



IDMEC/IST - Instituto de Engenharia Mecânica
INSTITUTO SUPERIOR TÉCNICO