Manual SPACAR95, software for dynamic analyses of multibody systems

A. L. Schwab, and J. P. Meijaard
Laboratory for Engineering Mechanics
Delft University of Technology
Mekelweg 2
NL-2628 CD Delft
The Netherlands
a.l.schwab@wbmt.tudelft.nl

November 2000

Contents

1. Introduction

This is a guide to the software package SPACAR95 for dynamic and kinematic analyses of multibody systems. Companion papers are [1] to [73] . Earlier versions were described in [25]. The software described in this document, SPACAR95 , is the sequential version of PLANAR [15] and SPACAR [25], the first 2 and 3 dimensional kinematic analyses programs based on the finite element method. Both were developed in the Laboratory for Engineering Mechanics at the Mechanical Engineering department of the Delft University of Technology. Compared to previous versions SPACAR95 has significant internal structural changes that are not apparent to the user. In addition it is somewhat more convenient to use, and it has a MATLAB user interface primarily for graphical representation of the results. The first author is much indebted to J.P.Meijaard for his inspiration, encouragement and support.

2. Purpose and objective.

SPACAR95 is a programming system for the analyses of motion of spatial multibody systems, including planar systems, with rigid and flexible links. In the development of the SPACAR95 system the main objective was to make use of finite element techniques. It is felt that a consistent description of the kinematics is a good starting point for the development of dynamic analyses techniques, for example vibration analyses of multibody systems. The finite element method approach satisfies automatically the requirement of a maximum of variety of multibody systems to be analyzed, provided that suitable finite elements are available. Typical available elements are truss, beam and hinge elements. In the case of more specialized analyzes one can think of belt and pulley, wheel and contact elements. Many multibody systems have multiple degrees of freedom. The system was designed such that multiple degrees of freedom could easily be treated. The program should serve a large class of users having different problems. All users have in common that they want to know the pure kinematical behaviour of the multibody system. The kinematical properties of the motion are specified by the discrete transfer functions which relate the various position parameters of the multibody system with the input motion. The SPACAR95 program system provides the following steps:

1. Definition of the multibody system (topology and geometry).
2. System preparation and initialisation.
3. Numeric integration of the equations of motion.

SPACAR95 contains possibilities for writing calculated results after each integration step to user defined ASCII files. The SPACAR95 system must be seen as a skeleton program forming the basis of the user's application. How the subroutines of the skeleton program are called and how the necessary input must be prepared is described in this manual.

3. Area of application.

Multibody systems find their application in numerous fields of engineering. In the case of multi degree of freedom systems having all elements rigid, or only a few deformable, SPACAR95 is at its best.

4. Characteristics.

As it is not possible to predict which questions will be asked about the multibody system in specific cases, a programming system, that is a toolbox, was developed. Consequently the user must compose his own program using the modules of the system. Pre and post-processing is not our cup of tea, the talents of the SPACAR95 system are in the field of mechanics. In a computer application three aspects can be distinguished: input definition, task definition and output definition. The SPACAR95 input defines the topological and geometrical composition of the multibody system to be analyzed. This input can be given in a format free ASCII input file with keywords and numbers. The SPACAR95 system is able to handle all multibody systems composed of truss, beam and hinge elements. The degrees of freedom can be rotational or translatory. The system is particularly suited for multidegree of freedom multibody systems having few deformable elements. The numeric integration of the equations of motion expressed in the degrees of freedom of the system is a standard task component. The calculation of the transfer function values is a sub task. The transfer function of order zero gives the mechanism position. The first and second order derivatives of the transfer functions with respect to the degrees of freedom, which are also calculated, are required for the determination of the velocities and accelerations. Other task components such as equilibrium forces for the nodes and the elements, linearized equations of motion etc. can be calculated on request of the user. A few output possibilities are incorporated in the system. Before and after having processed the input definition some statistics of the SPACAR95 system together with the given input definition of the to be analyzed multibody system are printed on the standard output. The system variables, such as position velocity acceleration force etc., can be written to an accordingly named ASCII file where each line stands for a discrete moment in time. Most post-processing programs such as MATLAB or grtool are capable of reading these files. The SPACAR95 system is written in FORTRAN77. We have tried to keep ourselves very strict to the standard. For those who want to program in FORTRAN and use SPACAR95 as a toolbox, please read the `credo` file. The SPACAR95 system does not call routines belonging to other systems, it is selfcontained and in that

5. Input Specification for SPACAR95

The input data is given in the form of a keyword with parameters. The keyword is a character string starting with an alphabetic character and with a maximum length of 8 characters. The parameters can be integer or real numbers and are seperated by white spaces. The real numbers can be given in exponent E-format. The end of line terminates the parameter list. In the description of the input the keywords are written in upper case characters but lower case or a mixture is allowed. I there is an asterisk character * on the first position of the line then this line will be treated as a comment line. Input can be given in blocks, where every block is terminated by the keyword `END`. After this command the analysis is started or continued. The total input is delimited by the physical end of file or by the end of file keyword `EOF`. The output is written to the console and can be captured by redirecting it to a file. After an analysis the input data and the current state is written as output. From this output a restart can be made.

The keywords of similar kind are listed in groups. Element numbers and node numbers are usually denoted by e and n which are integers bounded to the maximum number of elements and nodes. The numbering need not be consecutive. Node number p1 is an integer referring to the first position node of an element, where o1 refers to the first orientation node number.

5.1. Elements

`BEAM`	e	p1 o1 p2 o2	x y z	x y z is the initial direction of the principal y-axes of the cross-section of the beam.
`BEAMNL`	e	p1 o1 p2 o2	x y z	beam with bending corrected elongation, x y z is the initial direction of the principal y-axes of the cross-section of the beam.
`TRUSS`	e	p1 p2		
`HINGE`	e	o1 o2	x y z	x y z is the initial axes of rotation.
`BEARING`	e	p1 o1 p2 o2	x y z	x y z is the initial axes of rotation.
`SURFACE`	e	p1	c1..c10	c1..c10 are the coefficients of the quadratic surface.
`L2MIN1`	e	o1		the constraint on the Euler parameters, usually automatically created when the first time an Euler parameter is defined.
`PLBEAM`	e	p1 o1 p2 o2		planar beam.
`PLBEAMNL`	e	p1 o1 p2 o2		planar beam with bending corrected elongation.
`PLHINGE`	e	o1 o2		planar hinge.
`PLTRUSS`	e	p1 p2		planar truss.
`PLBELT`	e	p1 o1 p2 o2	r1 r2	planar belt with radii r1 and r2.
`PLBEAR`	e	p1 o1 p2 o2		planar bearing.
`PLWHEEL`	e	p1 o1 o2	r x y	planar wheel with radius r and initial axis of rotation x y, node o1 is the orientation of this axis and node o2 is the rotation along this axis.

Note 1: Nodes are implicitly defined by elements, the coordinate values are set to zero except for a 3-D orientation node where the Euler parameters are set to (1,0,0,0), resembling no rotation.

5.2. Nodes

`X`	n	x1 x2 x3 x4	any node n with coordinates x1 x2 x3 x4.
`XY`	n	x y	2D position node n with coordinates x y.
`BETA`	n	beta	2D orientation node n with coordinates beta.
`XYZ`	n	x y z	3D position node n with coordinates x y z.
`LAMBDA`	n	l0 l1 l2 l3	3D orientation node n with Euler parameters l0 l1 l2 l3.

Note 1: Since nodes are implicitly defined by elements, the non-zero coordinate values are usually specified with the any node keyword `X`.

5.3. Boundary Conditions

`FIX`	n	i	node n component i is fixed.
`INPUTX`	n	i	node n component i is input motion.
`DYNX`	n	i	node n component i is dynamic degree of freedom.
`RLSE`	e	i	element e deformation i is released.
`INPUTE`	e	i	element e deformation i is input motion.
`DYNE`	e	i	element e deformation i is dynamic degree of freedom.
`LINE`	e	i	element e deformation i is to be a linearization parameter.
`ENHC`	e1	i1 e2 i2	element e1 deformation i1 is nonholonomic, the corresponding kinematic coordinate is element e2 deformation i2.

Note 1: The implicit boundary condition on a node is a free node, whereas the implicit boundary condition on an element deformation is a fixed deformation.

Note 2: The kinematic coordinate must be a fixed deformation parameter.

5.4. Initial Values

`XD`	n	i	x	node n component i has velocity x.
`XDD`	n	i	x	node n component i has acceleration x.
`E`	e	i	x	element e deformation i has value x.
`ED`	e	i	x	element e deformation i has rate x.
`EDD`	e	i	x	element e deformation i has double rate x.
`TIME`	t			set the current time to t.
`DIR`	e		x y z	the first director of element e is x y z.
`LGEOMY`	e		x1..x13	element e general geometry values are x1..x13.
`LENGTH0`	e		l0	element e has initial length l0.

5.5. Dynamic Properties

`MASS`	n	m	concentrated mass m in XY and XYZ node n.
		Izz	moment of inertia Izz in BETA node.
		Ixx Ixy Ixz Iyy Iyz Izz	inertia tensor I values for LAMBDA node.
`EMASS`	e	rhoA	uniformly distributed mass rhoA per length for element e.
		rhoA a1 a2	uniformly distributed mass rhoA per length for PLBELT; in the initial state the amount of belt winded at pulley 1 measured from the contact point, counterclockwise, has a total length of a1*r1 and for pulley 2 measured from the contact point, clockwise, a length of a2*r2.
`ESTIFF`	e	EA GIp EIy EIz	stiffness parameters for BEAM element e
		EA	TRUSS.
		St	torsional stiffness for HINGE element.
		EA EI	PLBEAM.
		EA	PLTRUSS, PLBELT.
		St	PLHINGE.
		Sx Sy St	linear and torsional stiffness for PLBEAR element.
`EDAMPP`	e	x1..x4	damping values x1..x4 for element e, for the interpretation for different element types see ESTIFF.
`FORCE`	n	x1..x4	prescribed constant force x1..x4 at any node n.
`TORQUEL`	n	T	bodyfixed prescribed constant torque T at LAMBDA node n.

5.6. Settings and Controles

TIMESTEP	n	Tp	integrate the system with n number of constant output intervals over a time period of Tp.
HMAX	h		integrate the system with a maximum stepsize of h.
MAXITERAT	n		allow maximum n kinematic iterations per step.
EPSKIN	eps		kinematic convergence tolerance eps on the deformations.
EPSINT	eps		maximum local integration truncation error eps for the deformations.
EPSIND	eps		maximum local integration truncation error eps for the deformation rates.
FORCEMODE	0		no reaction forces, no element forces calculated.
	1		yes reaction forces, no element forces calculated.
	2		yes reaction forces, yes element forces calculated.
ANIMATE	0		graphic animation off.
	1		graphic animation on.
MATASC	0		no of the state in ASCII files.
	1		write at every output interval the state in ASCII formatted files which can be read by MATLAB.
JOBID	id		an integer identifier id which can be used in the program for indentification of a specific input file.
RESTART	0		continue numeric integrator.
	1		restart numeric integration.
EOF			a logical end of file marker, the input is terminated.
END			the reading of input is stopped and the analysis is continued until the end of the specified time interval is reached after which the program continues reading the input.

5.7. Example of an Input File

An example of an input file for the dynamic analysis of a slider-crank mechanism with an elastic connecting rod is:

```
PLBEAM 1 1 2 3 4
PLBEAM 2 3 5 6 7
PLBEAM 3 6 7 8 9
FIX 1
FIX 8 2
X 3 0.15 0.
X 6 0.30 0.
X 8 0.45 0.
INPUTX 2 1
DYNE 2 2
DYNE 2 3
DYNE 3 2
DYNE 3 3
TIMESTEP 100 5.0
MASS 8 0.033375
EMASS 1 0.2225
EMASS 2 0.2225
EMASS 3 0.2225
ESTIFF 2 0. 0.0012723
ESTIFF 3 0. 0.0012723
INPUTX 2 1
XD 2 1 1.5
END
```

And this is what the state should be after 5 seconds as written by the program:

```
PLBEAM     1    1    2    3    4
PLBEAM     2    3    5    6    7
PLBEAM     3    6    7    8    9
XY         1   0.000E+00   0.000E+00
BETA       2   7.500E+00
XY         3   5.199E-02   1.407E-01
BETA       4   7.500E+00
BETA       5  -4.443E-01
XY         6   1.863E-01   7.383E-02
BETA       7  -4.903E-01
XY         8   3.168E-01   0.000E+00
BETA       9  -5.279E-01
FIX        1   1
FIX        1   2
INPUTX     2   1
FIX        8   2
DYNE       2   2
DYNE       2   3
DYNE       3   2
DYNE       3   3
TIME      5.000E+00
XD         2   1  1.500E+00
XDD        2   1  0.000E+00
E          2   2  2.673E-03
ED         2   2 -3.049E-03
EDD        2   2 -6.476E-02
E          2   3  4.227E-03
ED         2   3 -5.511E-04
EDD        2   3 -2.260E-02
E          3   2  3.644E-03
ED         3   2 -4.904E-03
EDD        3   2 -1.124E-01
```

```
E          3   3  1.996E-03
ED         3   3 -5.660E-04
EDD        3   3 -1.237E-02
LENGTH0    1  1.500E-01
LENGTH0    2  1.500E-01
LENGTH0    3  1.500E-01
LGEOMY     1  0.000E+00
LGEOMY     2  0.000E+00
LGEOMY     3  0.000E+00
MASS       8   3.337E-02
EMASS      1  2.225E-01
EMASS      2  2.225E-01
EMASS      3  2.225E-01
ESTIFF     2  0.000E+00  1.272E-03
ESTIFF     3  0.000E+00  1.272E-03
HMAX      9.999E+03
MAXITERAT  10
EPSKIN    1.000E-05
EPSINT    1.000E-03
EPSIND    1.000E-03
FORCEMODE   1
ANIMATE    0
MATASC     0
RESTART    0
TIMESTEP 100  5.000E+00
END
```

6. Implementation of a New element Type

To implement a new element type there are three distinct things to do.

First, write a FORTRAN subroutine which calculates the element specific matrices as there are zero, first and second order derivates of the generalized deformations, initial geometry values from the current state, generalized stresses, mass matrix and convective force vector, third order derivative of the deformations, first order derivatives of the mass matrix and convective forces, material stiffness and damping matrix. The minimum required implementation is the zero, first and second order derivates of the generalized deformations, all other jobs may return zero. Work by example. For a 2-D element look at the planar beam element as implemented in the `PLBEAM` subroutine and for a new 3-D element type take the spatial beam subroutine `BEAM` as an example. The appropriate `JOB` identification can be found in the subroutine `LPROCS`.

Second, define the new element type in the subroutine `NLTDEF` by adding the element specific lines. If you want to add a 2-D element called `FOO` with 3 nodes, the first a `XY` node, the second a `BETA` node, and the third a `XY` node, and

2 deformation parameters and the first free element type identifier is 18 then you type:

```
NTLK(1)=2
NTLK(2)=1
NTLK(3)=2
CALL LTDEFI(18,'FOO',2,3,NTLK,IER)
```

Third, make the connection between the subroutine you have written, f.i. FOOBAR, and the element type identifier, 18, by adding the following line in the subroutine LPROCS:

```
ELSEIF(LTYPE.EQ.18)THEN
  CALL FOOBAR(JOB,KEL)
```

Check your dimensions with the current maximum implemented as can be found in the include file SPPARA which defines all parameters for the dimensions. Specific for a new element type are the parameters:

```
* MXLT  :MaXimum number of eLement Types.
* MXNL  :MaXimum number of Nodes per eLement.
* MXXL  :MaXimum number of X-components per eLement.
* MXEL  :MaXimum number of dEformations per eLement.
* MXML  :MaXimum number of Mass values per eLement.
* MXSL  :MaXimum number of Stiffness values per eLement.
```

7. Credo

```
*-purpose---------------------------------------------------------------
* Het vastleggen van de style van programmeren voor SPACAR95
*-edit log--------------------------------------------------------------
* 03-Feb-95 meijaard/schwab created.
*-----------------------------------------------------------------------

Houd u aan de FORTRAN77 standaard.
Een goed referentieboek is:
Metcalf,M.,"Effective Fortran 77",Oxford University Press,1989.
bv:     Alle F77 source in HOOFDLETTERS, commentaar strings en
    formats mogen in kleine letters.
    Gebruik DOUBLE PRECISION ipv REAL*8.
    Er bestaat maar een INTEGER type ipv INTEGER*2 of INTEGER*4.

Commentaar regels worden aangegeven met een * op de eerste positie
van de regel. Gebruik voor oude FORTRANIV source een C op de eerste
positie om duidelijk het verschil met F77 aan te geven.
Dit is ondermeer belangrijk bij het gebruik van lokale variabelen.
In FORTRANIV ging men ervan uit dat bij subsequentie aanroep van
```

een subroutine met lokale variabelen, de waarde van deze variabelen bewaard zou blijven. In F77 is dit NIET het geval. Wil men dit toch dan kan men dit bereiken door het SAVE statement toe te passen. We vermelden dit hier zo uitgebreid omdat het in het verleden een bron van fouten is geweest!

Houd u aan de impliciete declaratie:

```
      REAL (A-H,O-Z)
      INTEGER (I-N)
```

Declareer expliciet alle variabelen uit:
de parameterlijst van een SUBROUTINE of FUNCTION,
een COMMON block.
!en houd u aan de impliciete declaraties!.
Beter:
Declareer expliciet alle variabelen en compileer uw source met de optie IMPLICIT UNDEFINED. Dit voorkomt de beruchte fouten zoals:
LLX0 ipv LLXO.
!en houd u aan de impliciete declaraties!.

Spring 2 spaties in bij een DO-loop en een IF-THEN-ELSE block.

Bij ieder DO-loop hoort een afzonderlijk CONTINUE statement.

INTEGER constanten die mbv het PARAMETER statement gedefinieerd worden en die de grootte van array's aangeven bij voorkeur laten beginnen met de letters MX.

Het karakter voor het vervolgen van een statement op de volgende regel is een $ teken op positie 6.

Het label nr om een exceptie aan te geven, spring uit de loop etc, is 9999. Gebruik dit getal eventueel ook voor bijzondere INTEGER of REAL waarden. Op deze wijze vallen de bijzonderheden in de source in een oogopslag op. In de parameter file zijn hiervoor 3 constanten gedefinieerd: IUNDEF=9999,RUNDEF=9999.0 en CUNDEF='9999'.

SPACAR is een eindig elementen programma waarbij de begrippen node(knoop) en element erg vaak voorkomen.
Gebruik, bij voorkeur, voor een eLement-eigenschap de afkorting L en voor een Node-eigenschap de afkorting N.
bv: De coordinaten X van een eLement zijn XL(*)
    De coordinaten X van een Node zijn XN(*)

We gaan proberen SPACAR modulair op te bouwen en voorzien

vooralsnog 3 modulen:
1-SPKD: Kinematica + Dynamica.
2-SPID: Inverse Dynamica.
3-SPLN: Lineairisatie.
De basis module is SPKD. Alle andere modulen moeten hierop aansluiten.
Dus geen wijzigingen in SPKD.

Schrijf het FORMAT statements direct bij het bijbehorende WRITE statement. Gebruik afwijkende labelnummers, bv > 1000.

In een COMMON block is geen mix van CHARACTER en andere typen toegestaan.

Gebruik kleine letters voor filenamen.
Gebruik de extensie .f voor een FORTRAN source file.
Gebruik de extensie .h voor een PARAMETER include file.
Gebruik de extensie .h voor een COMMON block include file.

Wij ondersteunen het gebruik van het printer control character NIET. Laat echter het eerste karakter in iedere regel uitvoer een spatie zijn (FORMAT(1X,.....)).

De classificatie van de argumenten voor een SUBROUTINE en FUNCTION zijn:
I(nput): het argument MOET een waarde bij aanroep hebben en deze waarde wordt NIET veranderd.
O(utput): het argument moet een VARIABELE zijn en krijgt zijn waarde in de SUBROUTINE/FUNCTION.
I(nput)/O(utput): spreekt voor zich.

Assumed size array's altijd aangeven met een * (en geen 1 of zo).

Schrijf REAL getallen altijd met een .0 (bv 0 en 2 schrijven als 0.0 en 2.0).

Indien het nulstellen van een array in de "loop" zit (b.v. voor ieder integratiestapje of zo) gebruik dan nooit de routines CLRRA1,2,3 maar schrijf dit geheel uit mbv DO-loops.
De routines CLRRA1,2,3 vragen in verhouding erg veel tijd.
Is het nulstellen incidenteel, gebruik ze dan wel voor duidelijkheid.

A.L.Schwab

References

1. J.F. Besseling, 'The complete analogy between the matrix equations and the continuous field equations of structural analysis', in: *International Symposium on Analog and Digital Techniques Applied to Aeronautics, Liege 1963, Actes du colloque*, Bruxelles, 1964, pp. 223–242.

2. W. Visser, 'Berekening van drie-dimensionale staaf constructies, aangepast aan het gebruik van de electronische rekenmachine', Report LTM 269, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1964.

3. K. van der Werff, 'Grote verplaatsingen van vlakke staafconstructies toegepast op een kruisveerscharnier', Report LTM 408, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1968.

4. W. Visser, 'Algolprogramma voor berekening van grote verplaatsingen bij vlakke staafconstructies', Report LTM 411, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1968.

5. W. Visser and J.F. Besseling, 'Large displacement analysis of beams', Report WTHD 10, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, March 1969. (Compare with Report LTM 414.)

6. K. van der Werff, 'Codelijst en I/O-beschrijving van het programma "geometrie van de beweging van vlakke mechanismen"', Report LTM 513, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1973.

7. K. van der Werff, 'Numerieke berekening van de beweging van vlakke mechanismen', Report LTM 514, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1973.

8. J.F. Besseling, 'Non-linear analysis of structures by the finite element method as a supplement to a linear analysis', *Computer Methods in Applied Mechnics and Engineering* **3** (1974), pp. 173–194. (Also as Report WTHD 51, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1973.)

9. 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), pp. T3–T15.

10. J.F. Besseling, *Stijheid en sterkte 2, Toepassingen* (Mechanica 5), Oosthoek, Scheltema & Holkema, Utrecht, 1975.

11. L.F. Shampine and M.K. Gordon, *Computer Solution of Ordinary Differential Equations, the Initial Value Problem*, W.H. Freeman and Company, San Francisco, 1975.

12. K. van der Werff, 'Kinematics of coplanar mechanisms by digital computation', In *Proceedings of the Fourth World Congress on the Theory of Machines and Mechanisms* (September 8–13, 1975, Newcastle upon Tyne, England), pp. 685–690.

13. K. van der Werff, 'Dynamic analysis of planar mechanisms with rigid links', Report WTHD 85 and LTM 583, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1976.

14. 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), pp. 97–124. (Also as Report WTHD 94 and LTM 595, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1977.)

15. K. van der Werff, *Kinematic and Dynamic Analysis of Mechanisms, a Finite Element Approach*, (Dissertation), Delft University Press, Delft, 1977.

16. K. van der Werff, 'Kinematics and dynamics of n degree of freedom mechanisms', Report LTM 633, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1978.

17. K. van der Werff and H. Rankers, 'Getriebetyp-unabhängige Methode der Kinematik und Dynamik der Rader-Kurbelgetriebe', Report LTM 644, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1978.

18. J.F. Besseling, 'Finite element methods', in J.F. Besseling and A.M.A. van der Heijden (eds), *Trends in Solid Mechanics*, Delft University Press, Delft, 1979, pp. 53–78.

19. J.F. Besseling, L.J. Ernst, K. van der Werff, A.U. Koning, and E. Riks, 'Geometrical and physical nonlinearities: some developments in The Netherlands', *Computer Methods in Applied Mechanics and Engineering* **17/18** (1979), pp. 131–157. (Also as Report LTM 632, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1978.)

20. K. van der Werff, 'A finite element approach to kinematics and dynamics of mechanisms', in *Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms* (July 8–13, 1979, Montreal, Canada), The American Society of Mechanical Engineers, New York, 1979, pp. 587–590.

21. H. Rankers, B. van den Berg, A. van Dijk, A.J. Klein Breteler, and K. van der Werff, 'Computer aided design of mechanisms, the CADOM project of the Delft University of Technology', in *Proceedings of the Fifth World Congress on the Theory of Machines and Mechanisms* (July 8–13, 1979, Montreal, Canada), The American Society of Mechanical Engineers, New York, 1979, pp. 667–672.

22. J.F. Besseling et al., 'Applicatiecursus "Eindige Elementen Methode"', Report LTM 681, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1979.

23. J.F. Besseling, 'Derivatives of deformation parameters for bar elements and their use in buckling and postbuckling analysis', Report WTHD 94, Corrected version, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, July 1981.

24. 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), pp. 205–220. (Also as Report WTHD 143 and LTM 711, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1981.)

25. K. van der Werff, 'SPACAR, a programme system for the analysis of the motion of spatial mechanisms, user manual', Report LTM 739, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1983.

26. A.L. Schwab, 'Dynamica van mechanismen met vervormbare schakels' (Master's Thesis), Report LTM 758, Laboratory for Engineering Mechanics, Delft University of Technology, Delft 1983.

27. A.J. Klein Breteler, 'Movability and transfer quality of arbitrary mechanisms, an application of the FEM-theory', in J.S. Rao abd K.N. Gupta (eds) *Proceedings of the Sixth World Congress on the Theory of Machines and Mechanisms* (December 15–20, 1983, New Delhi, India), Wiley Eastern Limited, New Delhi, 1984, pp. 199–204.

28. A.J. Klein Breteler, 'The disc cam regarded as a special finite element in kinematics', in J.S. Rao abd K.N. Gupta (eds) *Proceedings of the Sixth World Congress on the Theory of Machines and Mechanisms* (December 15–20, 1983, New Delhi, India), Wiley Eastern Limited, New Delhi, 1984, pp. 1217–1220.

29. K. van der Werff, 'Finite element kinematical analysis of spatial mechanisms', in J.S. Rao abd K.N. Gupta (eds) *Proceedings of the Sixth World Congress on the Theory of Machines and Mechanisms* (December 15–20, 1983, New Delhi, India), Wiley Eastern Limited, New Delhi, 1984, pp. 362–368. (Also Report LTM 737, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1983.)

30. 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-Verlag, Berlin, 1984, pp. 381–400. (Also as Report LTM 767, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1983.)

31. J.B. Jonker, 'Dynamics of spatial mechanisms with flexible links', Report WTHD 171, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1984. (Also as Report LTM 804. Compare with Report LTM 794.)

32. J.F. Besseling, J.B. Jonker, and A.L. Schwab, 'Kinematics and dynamics of mechanisms', *Delft Progress Report* **10** (1985), pp. 160–172. (Also as Report LTM 799, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1985.)

33. B. Tanuwidjaja, 'Simulation des dynamischen Verhaltens ebener Mechanismen mit spielbehafteten Drehgelenken mit Hilfe der Finite-elementemethode', Report WTHD 173, Section Production Automation and Theory of Mechanisms, Delft University of Technology, Delft, 1985.

34. A.L. Schwab, 'The wheel in 3D and other special finite elements for kinematic and dynamic analyses of multy body systems', Report LTM 820, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1985.

35. J.F. Bessseling, 'Large rotations in problems of structural mechanics', in P.G. Bergan, K.-J. Bathe, and W. Wunderlich (eds), *Finite Element Methods for Nonlinear Problems* (Europe-US Symposium, Trondheim, Norway 1985), Springer-Verlag, Berlin, 1986, pp. 25–39. (Also as Report LTM 810, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1985.)

36. J.B. Jonker, 'Dynamics of active mechanisms with flexible links', in G. Bianchi and W.O. Schiehlen (eds), *Dynamics of Multibody Systems*, Springer-Verlag, Berlin, 1986, pp. 103–118. (Also as Report LTM 812, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1985.)

37. J.B. Jonker, 'A computer-oriented method for linearization of the dynamic mechanism equations', Report LTM 822, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.

38. C.M. Wouters, 'Verification of simulation models of a hydraulically actuated servomechanism' (Master's Thesis), Report LTM 833, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.

39. J.B. Jonker, 'Dynamic simulation of of robotic manipilators with flexible links', in H. van Brussel (ed.), *Proceedings 16th International Symposium on Industrial Robots, 8th International Conference on Industrial Robot Technology*, Springer-Verlag, Berlin, 1986, pp. 365–376. (Also as Report LTM 834, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.)

40. R.A.G. van Tol, 'Linearisering van de bewegingsvergelijkingen van ruimtelijke mechanismen in een voorgeschreven nominale trajectorie' (Master's Thesis), Report LTM 839, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.

41. R.A.G. van Tol, 'Robot simulatie met een stuksgewijs gelineariseerde inverse model regeling, simulatie van een "Experimenteel Robotachtig Proces" (ERP) bij het volgen van een voorgeschreven baan' (Master's Thesis), Report LTM 841, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.

42. P.M.R. Wortelboer, 'Inverse dynamics of manipulators–implementation in the finite element/mechanism package SPACAR' (Master's Thesis), Report LTM 842, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1986.

43. J.F. Besseling, 'Some non-linear problems for evaluating FEM systems', in *Quality Assurance in FEM Technology* (Proceedings of the Fifth World Congress, Salzburg, Austria, 5–9 October 1987), Robinson and Associates, Bridgestowe, 1987, pp. 43–52. (Also as Report LTM 847, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1987.)

44. R.B.M. Bosman, 'De consistente massabeschrijving van een balkelement dat grote rotaties kan ondergaan' (Master's Thesis), Report LTM 856, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1987.

45. H.J. van Oostveen, 'Simulatie van de beweging van een dynamische bandbuffer' (Master's Thesis), Report LTM 866, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1987.

46. P. van Turennout, 'Modelling and control of a flexible satellite using the finite element package SPACAR', Report A87.026, Control Engineering Laboratory, Delft University of Technology, Delft, 1987.

47. J.B. Jonker and C. Keus, 'A finite element dynamic analysis of robotic manipulators with flexible links', in J.F. Besseling and W. Eckhaus (eds), *Trends in Applications of Mathematics to Mechanics*, Springer-Verlag, Berlin, 1988, pp. 160–171. (Also as Report LTM 868, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1987.)

48. C. Keus, 'End-point position control of flexible link manipulators' (Master's Thesis), Report A-434, Laboratory for Measurement and Control, Delft University of Technology, Delft, 1988.

49. J.B. Jonker et al., 'SPACAR user manual and system description', Report LTM 872, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1988.

50. J.B. Jonker, *A Finite Element Dynamic Analysis of Flexible Spatial Mechanisms and Manipulators* (Dissertation), Delft University of Technology, Delft, 1988. (Also as Report LTM 873.)

51. J.P. Meijaard, 'Numerieke integratie van gewone differentiaalvergelijkingen met behulp van Runge-Kutta-methoden', Report LTM 876, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1988.

52. L.A. van Wijk, 'Het berekenen van steady-state oplossingen van de bewegingsvergelijkingen van mechanismen' (Master's Thesis), Report LTM 887, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1988.

53. J.B. Jonker, 'A finite element dynamic analysis of spatial mechanisms with flexible links', *Computer Methods in Applied Mechanics and Engineering* **76** (1989), pp. 17–40.

54. J.B. Jonker and J.P. Meijaard, 'SPACAR—computer program for dynamic analysis of flexible spatial mechanisms and manipulators', in W.O. Schiehlen (ed.), *Multibody Systems Handbook*, Springer-Verlag, Berlin, 1990, pp. 123–143. (Also as Report LTM 893, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1989.)

55. J.B. Jonker, 'A finite element dynamic analysis of flexible manipulators', *International Journal of Robotics Research* **9**(4) (1990), pp. 59–74.

56. J.P. Meijaard, 'A comparison of numerical integration methods with a view to fast simulation of mechanical dynamical systems', in E.J. Haug and R.C. Deyo (eds), *Real-time Integration Methods for Mechanical System Simulation*, Springer-Verlag, Berlin, 1991, pp. 329–343. (Also as Report LTM 901, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1989.)

57. J.B. Jonker, 'Linearization of dynamic equations of flexible mechanisms—a finite element approach', *International Journal for Numerical Methods in Engineering* **31** (1991), pp. 1375–1392.

58. 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), pp. 1691–1710. (Also as Report LTM 930, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1990.)

59. G.M. Pronk, *The shoulder girdle, analysed and modelled kinematically,* (Dissertation), Delft University of Technology, Delft, 1991.

60. F.C.T. van der Helm, *The shoulder mechanism, a dynamic approach,* (Dissertation), Delft University of Technology, Delft, 1991.

61. J.F. Besseling and D.G. Gong, 'Numerical simulation of spatial mechanisms and manipulators with flexible links', in A.A.O. Tay and K.Y. Lam (eds), *Computational Methods in Engineering, Advances & Applications*, World Scientific, Singapore, 1992, pp. 407–412.

62. A.J. van Soest, A.L. Schwab, M.F. Bobbert and G.J. van Ingen Schenau, 'SPACAR: a software subroutine package for simulation of the behaviour of biomechanical systems', *Journal of Biomechanics* **25** (1992), pp. 1219–1226.

63. D.G. Gong, 'Simulation of machinery systems including dynamic properties of the electrical drivers', Report LTM 1018, Laboratory for Engineering Mechancis, Delft University of Technology, Delft, 1993.

64. A.J. van Soest, A.L. Schwab, M.F. Bobbert and G.J. van Ingen Schenau, 'The influence of the biarticularity of the gastrocnemius muscle on vertical–jumping achievement', *Journal of Biomechanics* **26** (1993), pp. 1–8.

65. D.G. Gong, 'A beam element for rigid body systems', Report LTM 1027, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1994.

66. P. Ritmeijer, 'Het kinematisch en dynamisch gedrag van een bouwrobot' (Master's Thesis), Report LTM 1028, Laboratory for Engineering Mechanics, Delft University of Technology, Delft, 1994.

67. J.P. Meijaard, 'Direct determination of periodic solutions of mechancal dynamical systems', *Archive of Applied Mechanics* **64** (1994), pp. 249–257.

68. F.C.T. van der Helm, 'Analysis of the kinematic and dynamic behaviour of the shoulder mechanism', *Journal of Biomechanics* **27** (1994), pp. 527–550.

69. F.C.T. van der Helm, 'A finite element musculoskeletal model of the shoulder mechanism', *Journal of Biomechanics* **27** (1994), pp. 551–569.

70. D.G. Gong, *Numerical Simulation of Mechanisms and Manipulator Robots with Flexible Links and Flexible Joints* (Dissertation), Delft University of Technology, Delft, 1995.

71. R. Happee and F.C.T. van der Helm, 'The control of shoulder muscles during goal directed movements, an inverse dynamic analysis', *Journal of Biomechanics* **28** (1995), pp. 1179–1191.

72. J.P. Meijaard, 'Validation of flexible beam elements in dynamics programs', *Nonlinear Dynamics* **9** (1996), pp. 21–36.

73. F.C.T. van der Helm and H.E.J. Veeger, 'Quasi-static analysis of muscle forces in the shoulder mechanism during wheelchair propulsion', *Journal of Biomechanics* **29** (1996), pp. 39–52.