J. Biomechanics Vol. 25, No. 10, pp. 1219–1226, 1992.
Printed in Great Britain

0021–9290/92 $5.00 + .00
Pergamon Press Ltd

TECHNICAL NOTE

SPACAR: A SOFTWARE SUBROUTINE PACKAGE FOR SIMULATION OF THE BEHAVIOR OF BIOMECHANICAL SYSTEMS

ARTHUR J. VAN SOEST,*‡ AREND L. SCHWAB,† MAARTEN F. BOBBERT* and GERRIT JAN VAN INGEN SCHENAU*

*Faculty of Human Movement Sciences, Vrije University, Amsterdam, The Netherlands and †Delft University of Technology, Faculty of Mechanical Engineering, Mekelweg 2, NL 2628 CD Delft, The Netherlands

Abstract—Direct dynamics computer simulation is gaining importance as a research tool in the biomechanical study of complex human movements. Therefore, the need for general-purpose software packages with which the equations of motion can be derived automatically and solved numerically is growing. In this paper such a method is described: SPACAR. The method is compared to well-known commercially available software packages. On the basis of the results obtained on a test problem simulated with both SPACAR and DADS, it is concluded that both methods are accurate; DADS is much faster. The user-friendliness of SPACAR is less than that of DADS. However, SPACAR has two major advantages. First is the basic deformability of all elements, which allows handling of all kinds of problems within a unified framework; second is the full availability of the source code, which allows the experienced user to broaden the scope of possibilities to any extent.

INTRODUCTION

A large part of biomechanical research is aimed at understanding the organization of gross human movements. Such movements involve a number of body segments that interact with each other and with the environment. In recent years, computer simulation of such movements is performed more and more often. It is felt by many researchers that direct dynamics computer simulation is a valuable tool to increase the depth of understanding of complex human movements (Hatze, 1981; Zajac and Winters, 1990). Unfortunately, the formulation and numerical solution of the equations of motion of a multi-body mechanical system is not a simple task. In our view, this fact should not restrict the use of direct dynamics simulation as a research tool to those researchers who can solve the theoretical and numerical problems involved. In other words, a growing demand exists for general-purpose software packages: such packages allow the researcher to focus on his biomechanical problem without being cluttered up with numerical methods.

In the past 20 years, a substantial amount of work has been done on the formulation of generally applicable computer-oriented methods for the derivation and solution of the equations of motion of three-dimensional multi-body systems. The state of the art in multi-body dynamics computer software is reviewed by Haug (1984, 1989) and Schielen (1990). At present a number of general-purpose software packages is commercially available; the most well-known are ADAMS ('Automated Dynamical Analysis of Mechanical Systems', Mechanical Dynamics Inc., Ann Arbor) and DADS ('Dynamical Analysis and Design of Systems', Computer Aided Design Software Inc., Oakdale). These methods allow both rigid and deformable elements, which may be connected in both open and closed loops; the equations of motion are derived numerically using absolute coordinates and a Newton–Euler approach. Both software packages are applicable to a wide class of problems; both are characterized by a user-friendly interface. Nevertheless, recently significant shortcomings of DADS (when applied in the field of biomechanics) were noted by van den Bogert (1990).

The aim of this paper is to describe a less well-known general-purpose method for generation and numerical solution of the equations of motion of multi-body mechanical systems: SPACAR. SPACAR resembles DADS and ADAMS in its general characteristics. However, it differs considerably in both theoretical basis and development/distribution philosophy. Therefore, we expect that its description is interesting to biomechanists involved in direct dynamics simulation.

In our view, the following criteria (presented in order of importance) should be used when comparing general-purpose software packages for direct dynamics simulation:

—Flexibility: although we only consider general-purpose packages, this does not mean that, using such packages, everything is possible [see van den Bogert (1990) for examples]. Furthermore, because of commercial interests, the user is in many cases not given opportunity to adapt the software himself.

—Accuracy: the system of differential equations governing the movement of the human body is in many instances unstable. Therefore, local numerical accuracy must be very high and extreme care must be taken when applying numerical integration.

—User-friendliness: although, due to the complexity of the subject, a considerable time investment from the software user may be expected, user-friendliness is an important characteristic. Especially, flexibility in the definition of the model and the boundary conditions as well as in output data management is important.

—Calculation speed and numerical efficiency: the importance of these depends strongly on the complexity of the models used, the number of simulations to be performed

Received in final form 18 *February* 1992.

‡Author to whom correspondence should be addressed at: Department of Functional Anatomy, Faculty of Human Movement Sciences, van der Boechorststraat 9, NL 1081 BT Amsterdam, The Netherlands.

(which can be high when, for example, dynamic optimization is considered) and the available hardware.

In this paper, first a global description of SPACAR is given; next, SPACAR is compared with DADS and ADAMS using the criteria mentioned above. In part, this comparison is based on simulations of the behavior of an example model, which were performed with both SPACAR and DADS.

THE SPACAR APPROACH

SPACAR is a computer subroutine package (written in FORTRAN 77) for the analysis of mechanical engineering problems, developed at the Technical University of Delft (van der Werff, 1977; Schwab, 1983). With SPACAR, the behavior of systems of two-dimensional/three-dimensional deformable/rigid bodies, connected in open/closed loops, can be simulated. SPACAR is based on the finite element method, which implies a discrete approximation to a continuous problem. The finite element method is characterized by absolute coordinates, a Eulerian approach and numerical equations. Thus, in its general characteristics SPACAR is comparable to commercially available methods such as DADS and ADAMS.

The main text of this article provides a global description of the method only; for a mathematical description of the method, the reader is referred to the Appendix.

In SPACAR, the direct dynamics problem is split into two distinct parts:

(1) Kinematics: calculation of the position, velocity and acceleration of the entire system given the position, velocity and acceleration of a subset of the system variables. This subset of variables that can be chosen freely by the user is referred to as the degrees-of-freedom (D_f).

(2) Dynamics: calculation of the accelerations of the D_f, given all positions, velocities and forces.

In SPACAR a model is constructed by assembling different types of elements, just like using a Meccano construction kit. As an example, we will construct a planar model of the skeletal system of a human being performing a two-legged vertical jump. The actual system [Fig. 1(a)] can be described by four rigid segments: feet, lower legs, upper legs and head-arms-trunk (HAT). These segments are connected in three joints: ankle, knee and hip. One extra 'joint' is introduced between the feet and the ground to describe the orientation of the entire system [Fig. 1(b)]. Anatomically, this joint would correspond to the metatarsophalangeal (MTP) joint.

Each segment is modelled using a BEAM element. This type of element has three independent modes of deformation: elongation of the element, bending at one end point, and bending at the other end point. Obviously, the position of any rigid element in two dimensions can be described by three coordinates. For every independent deformation that is introduced, one extra coordinate is needed to describe this deformation. Thus, the planar BEAM, having three deformations, must be characterized by $3+3=6$ independent coordinates: x, y and β of each end point [see Fig. 1(c)]. In cases where rigid BEAMS are needed, the deformations are simply set to zero. Thus, a rigid BEAM is described by six coordinates and three constraint equations specifying that the deformations equal zero. Note that the number of D_f of an element not fixed to the reference frame (and, for that matter, of a multilink system of elements), equals its number of coordinates minus the number of deformations defined to equal zero.

Each joint is modelled using a HINGE element [see Fig. 1(d)]. This type of element has the torsion in the HINGE as its only deformation. Interestingly, the HINGE is defined in such a way, that its position is not defined; it only has an orientation. As this orientation is described by one coordinate, and as it has one mode of deformation, in total the HINGE element can be characterized by two nodes, each having one β coordinate.

In total, the finite element model of the jumper consists of eight elements: four BEAMS and four HINGES [see Fig. 1(b)]. The connectivity of the model is obtained by giving elements nodes in common. For example, the coordinates of the proximal end of the lower legs are identical to the coordinates of the distal end of the upper legs.

Kinematics

When all eight elements are undeformable, the entire system is actually one rigid body. Thus, in case it is not in any way fixed to the reference frame, it should have three degrees-of-freedom (D_f). This can be checked by taking the difference between the total number of coordinates and the total number of deformations (since all deformations are now defined to be zero, see above). In this case, 5×2 (x, y of BEAM end points) $+4\times2$ (β of BEAMS, shared with HINGES) $+1$ (β of 'free' end of HINGE at the MTP joint) $=19$ coordinates minus 4×3 (BEAM) $+4\times1$ (HINGE) $=16$ deformations indeed equals $3D_f$. The system can be fixed in two-dimensional space by prescribing the coordinates of the toe and the orientation of the 'free' end of the HINGE at the MTP joint.

In SPACAR, any subset of the defined coordinates and/or deformations can be chosen as D_f, as long as the elements of this subset are independent and the number of elements equals the number of D_f. For our example, it is convenient to choose the deformations of the four HINGE elements as D_f; this subset of the set of deformations of the mechanism is denoted by $\mathbf{E}_m$. By definition, all coordinates are functions of D_f: $\mathbf{X}=F(\mathbf{E}_m)$ where $\mathbf{X}$ is the vector of coordinates and F is

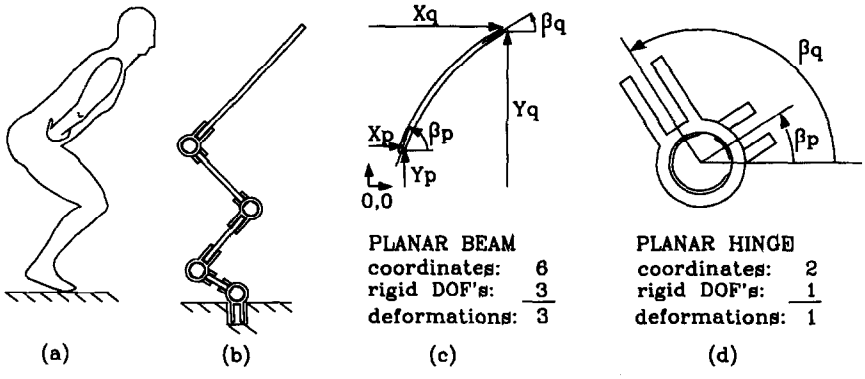

Fig. 1. (a) Human subject performing vertical jump; (b) finite element method model of the human subject; (c) description of the BEAM element; (d) description of the HINGE element.

the function relating $\mathbf{X}$ to $\mathbf{E}_m$ [see equation (A1)]. Unfortunately, this (highly nonlinear) function F cannot be derived in a standard way. However, given a (initial) position of the entire system, the first and second partial derivatives of all coordinates with respect to all D_f can be obtained in a standard way and are calculated in SPACAR:

$$DF = \partial \mathbf{X}/\partial \mathbf{E}_m \text{ and } D^2F = \partial^2 \mathbf{X}/\partial \mathbf{E}_m^2,$$

where D and D^2 symbolically represent the first and second differential operators [see equations (A5)–(A8)]. These matrices of instantaneous values of partial derivatives, sometimes referred to as Jacobians, are the first- and second-order transfer functions. The first-order transfer function gives (in the current position) the ratio between the change in the coordinates and a (small) change in D_f. Note that these ratios (partial derivatives) depend strongly on position. The zero-order transfer function, F in $\mathbf{X} = F(\mathbf{E}_m)$, cannot be calculated directly; in SPACAR it is obtained with help of the initial position and the first- and second-order transfer functions using an iteration scheme [see equation (A13)].

Given the motion in the D_f and the first- and second-order transfer functions, the motion of the entire system is determined; therefore, the transfer functions contain all the kinematic information of the system.

Dynamics

In a direct dynamics analysis, the accelerations of the D_f are to be calculated, given the model constitution and the position and velocity of the D_f. In SPACAR this is done as follows. First, the position and velocity of the entire system are calculated from those of the D_f with help of the transfer functions in the way described above. Next, all active forces are calculated, which is possible since, in general, forces are functions of time, position and/or velocity. Again using the first-order transfer function, these forces can be mapped into the D_f force space. The mass matrix is mapped in the same way. Using the d'Alemberts' principle, a system of linear equations results that can be symbolically represented as:

$$\mathbf{M}\ddot{\mathbf{E}}_m = \mathbf{F},$$

where $\mathbf{M}$ is the (mapped) mass matrix, $\ddot{\mathbf{E}}_m$ is a vector of the accelerations of the D_f (in our example all D_f are deformations) and $\mathbf{F}$ is a vector of (mapped) active forces [see equation (A15)]. This system of equations can be solved for $\ddot{\mathbf{E}}_m$, the acceleration of the D_f [see equation (A16)]. Subsequent positions and velocities of the D_f are obtained by numerical integration. The reader is referred to the Appendix for a more detailed description of the mathematics involved.

Using the same approach, inverse dynamic problems can be solved as well: in that case, the position, velocity and acceleration of the system at each instant of time are given, and the forces to be delivered by the D_f are calculated. In such applications, there is no need for numerical integration, as the movement is already known. One of the main problems in inverse dynamics is how to numerically differentiate the usually noisy recorded positional data in order to obtain accurate velocity and acceleration data.

COMPARISON OF SPACAR WITH EXISTING SOFTWARE PACKAGES

Although, as stated above, SPACAR, ADAMS and DADS are comparable in their general characteristics, a number of differences exist between SPACAR and the other methods mentioned. These differences are largely related to the fact that the developers of SPACAR were driven mostly by academic interest. Thus, at present SPACAR is nothing more than a set of subroutines, the source code of which is fully available to SPACAR-users. Readers interested in obtaining SPACAR should contact the second author of the present paper.

Here, SPACAR is compared to DADS. This comparison is restricted to two-dimensional applications of both methods. For an example of a three-dimensional application of SPACAR, see van der Helm (1991), who succeeded in modelling the highly complex kinematics of the shoulder girdle. The comparison made here is based on the criteria formulated in the introduction.

Flexibility. Recently a number of limitations of DADS were enumerated by van den Bogert (1990), who used DADS extensively in two-dimensional simulations of horse walking; to our knowledge, these problems apply to ADAMS as well. The limitations are

—adding or removing constraints during simulation is impossible; this would be necessary to create a 'hard' connection between system and environment;

—differential equations cannot be used in 'user force' routines; thus, no straightforward method is available to drive a model by muscles whose behavior is governed by ordinary differential equations (ODEs);

—a stiff integration algorithm cannot be substituted for the standard algorithm.

Furthermore, in DADS no straightforward method is available to define conditions under which the simulation is to be terminated. Finally, embedding DADS in optimization software is not straightforward (van Soest and van den Bogert, 1991).

In the standard version of SPACAR, constraint manipulation and the use of 'additional' ODEs was not possible. As the need of such facilities arose, a subroutine was developed for constraint manipulation [see van Soest *et al.* (1992)]; furthermore, theory has been developed concerning impulse equations which allow correct handling of impact (Schwab, personal communication). Also, a straightforward method for definition of 'additional' ODEs that are integrated in parallel with the mechanical system was created by the first author of the present paper. An example of an application in which such additional ODEs are used is given by van Soest *et al.* (1992). Finally, the subroutine containing the integration algorithm could be readily replaced in the standard version. Definition of conditions of termination, as well as the embedding in optimization software, are straightforward in SPACAR.

Accuracy. By stating that methods should be accurate, we mean that if no constraints are imposed on calculation time, the correct solution should be obtained. Formally, for any but the simplest models, it is not possible to prove that a solution is correct. However, when different methods yield identical solutions, this is a strong indication that these solutions are correct. When the accuracy of DADS and SPACAR are judged on this basis (from the numerical example presented below), it is concluded that both methods are accurate. It must be noted that although local accuracy in integration is specified by the user in the algorithms used, global accuracy depends on the stability of the system of ODEs. This stability is essentially a model property. Although, in general, numerical integration is not advised for unstable systems (Shampine and Gordon, 1975), the results obtained here as well as elsewhere (van Soest *et al.*, 1992) show that when care is taken, it is possible to obtain accurate results for unstable systems.

An interesting difference between SPACAR and DADS that might affect accuracy is that in SPACAR the coordinates used as D_f form a fixed subset specified by the user, whereas in DADS the D_f are selected (and possibly changed) by the software. Therefore, SPACAR requires more insight from the user. In DADS, the user is protected from making a poor choice; as a consequence, some time is spent by the software on monitoring and possibly changing the D_f. On the other hand, problems due to the occurrence of kinematic singularities are solved by changing the D_f in DADS, whereas in SPACAR kinematic singularities may lead to a crash. In practice, however, we have never experienced a crash of SPACAR.

User-friendliness. As stated before, SPACAR is nothing more than a set of subroutines. No attention was given to development of a user-friendly interface in SPACAR. Thus, in SPACAR a straightforward way for model definition is available and limited flexibility is present in both graphical and numerical output organization. In fact, any user of SPACAR is expected to adjust relevant subroutines to his particular situation. In contrast, both ADAMS and DADS are characterized by a high degree of user-friendliness, including pre- and post-processors.

Calculation speed and numerical efficiency. In order to assess calculation speed and numerical efficiency of both SPACAR and DADS, a test model was created. The behavior of this test model was simulated with both methods. Model description and simulation results are given below. From the results obtained with this model, it can be concluded that a large difference in calculation speed exists between DADS and SPACAR. At a working accuracy (approximately five correct digits in the angular accelerations) calculation times were 10 and 202 s for DADS and SPACAR, respectively. A priori, it was expected that DADS would be superior to SPACAR with respect to calculation speed: due to the underlying theory in SPACAR extra variables related to the deformations of the bodies are introduced, even when these deformations are in fact not used. However, it was not expected that speed ratio would be of the order of 1:20. A large part of this difference in speed probably results from the fact that the developers of SPACAR paid more attention to theoretical aspects than to numerical efficiency in developing its present version. Significant improvements in calculation speed have been established by substitution of more efficient linear algebraic algorithms (van der Helm, personal communication).

It must be stated that the comparison made here is based on a two-dimensional model. Modelling and simulation in three dimensions contain extra problems of such significance that, in our view, the comparison made here should not be extrapolated to applications in three dimensions.

NUMERICAL EXAMPLE: COMPARISON OF SPACAR AND DADS

A comparison of calculation speed and accuracy between SPACAR and DADS was made using the skeletal model developed above [Fig. 1(b)]. Using this model, it was intended to perform simulations of vertical jumping. Vertical jumping was chosen because it was expected that this is an unstable task, hence differences might result in the solutions obtained with these methods. In order to perform a direct dynamics simulation of vertical jumping, the model was equipped with simple moment actuators in hip, knee and ankle joints. After some experimentation, it was decided to use constant joint moments, primarily because constant joint moments were found to result in a high-degree of instability of the system. The values of these joint moments were chosen in such a way that a coordinated jump resulted. Joint angles (between 0 and π radians) were defined as follows: an angle of zero indicates full flexion at the hip and the knee and full dorsiflexion at the ankle; the toe angle was defined as the smallest angle between the feet and the horizontal. Segmental parameter values of the model are shown in Table 1. Joint moment values and starting conditions of the simulations (angles and angular velocities in the joints), are presented in Table 2. A value of 9.81 was used for gravitational acceleration.

In this simulation the heel is never in contact with the ground. Given this fact, and modelling ground contact at the toes as a frictionless hinge joint, the model has four D_f until the instant of takeoff. As only the push-off phase will be simulated in this example, there is no need to add or remove constraints during the simulation.

Table 1. Segmental parameters. Icm is moment of inertia, relative to the segment's center of mass. CMPROXREL is position of the segment's center of mass, expressed as a fraction of segment length, measured from the proximal end of the segment

	Length (m)	Mass (kg)	Icm (kg m^2)	CMPROXREL
Foot	0.16	2.0	0.01	0.4
Lower leg	0.40	6.0	0.10	0.4
Upper leg	0.44	14.0	0.30	0.4
Trunk	0.82	45.0	2.50	0.4

Table 2. Joint parameters. MTP indicates metatarsophalangeal joint. $\varphi_{t=0}$ indicates joint angle at the start of the simulation. Initial joint angular velocity was 0.0 for all joints

	Joint moment (N m)	$\varphi_{t=0}$ (rad)
MTP	0	0.6
Ankle	170	1.5
Knee	144	1.5
Hip	260	1.4

The behavior of this model was simulated using both SPACAR and DADS, in order to check if identical results were obtained and in order to get an impression of the trade-off between global accuracy and calculation time. Calculations were performed on an Apollo DN 2500 computer, which runs at approximately 0.5 Mflops.

Two types of error may occur when performing direct dynamics simulations with DADS or SPACAR. The first type of error concerns the algebraic constraint equations, that are solved iteratively. These constraint equations are formulated in terms of nodal coordinates in DADS and in terms of element deformations in SPACAR [equation (A13)]. The second type of error concerns the numerical integration of the state equations. In DADS and SPACAR, both types of error are locally controlled by the user. The global effects of the allowed local errors depend, of course, on the stability of the state equations, which is a model property.

For DADS, the first type of accuracy did not have an influence for any reasonable value of the parameter involved. For SPACAR, this accuracy was found not to influence the obtained solution, as long as the required accuracy was higher than the integration accuracy. Integration algorithms used in both methods are based on Shampine and Gordon (1975); these variable-order, variable-stepsize Adams–Bashford predictor/Adams–Moulton corrector algorithms are characterized by control of the local integration error up to a user-defined accuracy. As a first step, this local accuracy was increased by factors of 10 until no further change in model behavior could be obtained. The solutions converged for both DADS and SPACAR. The final 'reference' solutions thus obtained with both methods were extremely close to each other. Time histories of the joint angles are shown in Fig. 2. When presented graphically, final solutions obtained with SPACAR and DADS cannot be discerned. As can be seen from this figure, the behavior is unstable, which makes this model a severe test for numerical methods such as SPACAR and DADS.

Next, the trade-off between calculation speed and global accuracy was studied. Global accuracy was defined here in terms of the difference in angular accelerations between the actual solution and the reference solution. In fact, the sum of

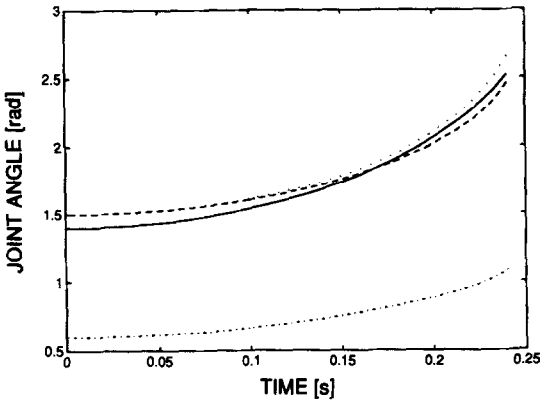

Fig. 2. Simulated joint angles for hip (——), knee (·····), ankle (– – –) and toe (–·–·–·) vs time for the test model as described in the text.

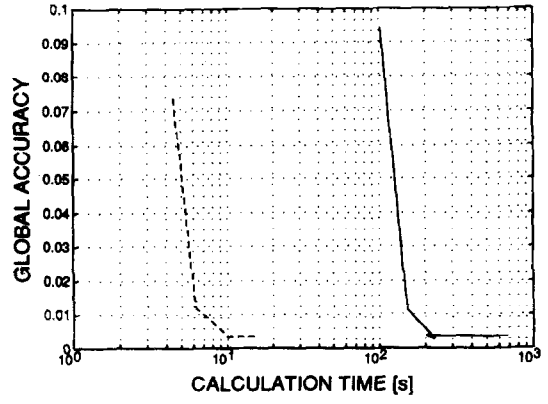

Fig. 3. Global accuracy vs calculation time for DADS (– – –) and SPACAR (— —) in a typical two-dimensional model. See text for details.

the square of these differences at $t=0.24$, which is close to takeoff, was used. Global accuracy was manipulated by substituting different values for the parameter defining the required local accuracy in numerical integration. Plots of global accuracy vs calculation time are presented in Fig. 3 for both SPACAR and DADS. From this figure it can be seen that DADS is approximately 20 times faster than SPACAR at any level of global accuracy.

CONCLUSION

The global characteristics of SPACAR are comparable to those of commercially available methods such as DADS and ADAMS. SPACAR has two major advantages. The first concerns the theoretical basis, i.e. the basic deformability of all elements, which allows handling of all kinds of problems within a unified framework. A second advantage is the availability of the complete source code, allowing the experienced user to tune the software to his specific demands and to broaden the scope of possibilities in cases where the standard version is too restrictive. For the inexperienced and occasional user, the absence of a user-friendly interface in SPACAR is a disadvantage. It was shown for a typical two-dimensional model that both SPACAR and DADS yield accurate results. With respect to calculation speed, DADS was found to be far superior to SPACAR. The present comparison was made on the basis of a two-dimensional example. When direct dynamics simulations of large three-dimensional models are considered, another check and comparison seem worthwhile.

Acknowledgement—The authors gratefully acknowledge the help of Dr A. J. van den Bogert, Department of Veterinary Anatomy, University of Utrecht, The Netherlands, who took part in discussions on the formulation of the test model used in this study, and who performed the DADS simulations of the behavior of this model.

REFERENCES

Bogert, A. J. van den (1990) Musculoskeletal modelling: the DADS experience. *ISB Newsletter* **39**, 4–6.

Hatze, H. (1981) *Myocybernetic Control Models of Skeletal Muscle*. University of South Africa, Pretoria.

Haug, E. J. (1984) *Elements and Methods of Computational Dynamics*, NATO advanced study inst. on computer aided analysis and optimization of mechanical system dynamics, pp. 3–36. Springer, Berlin.

Haug, E. J. (1989) *Computer Aided Kinematics and Dynamics of Mechanical Systems, Vol. 1: Basic Methods*. Allyn and Bacon, Boston.

Helm, F. C. T. van der (1991) The shoulder mechanism: a dynamic approach. Ph.D. thesis, pp. 85–128, Delft University of Technology, Delft.

Meriam, J. L. (1975) *Statics* (S. I. version), pp. 269–317. Wiley, New York.

Schielen, W. (1990) *Multibody Systems Handbook*. Springer, Berlin.

Schwab, A. L. (1983) Dynamica van mechanismen met vervormbare schakels. Lab. Engng Mech. report TM 758. Delft University of Technology, Delft (in Dutch).

Shampine, L. F. and Gordon, M. K. (1975) *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*. W. H. Freeman, San Francisco.

Soest, A. J. van, Schwab, A. L., Bobbert, M. F. and Ingen Schenau, G. J. van (1992) The influence of the biarticularity of the gastrocnemius muscle on vertical jumping achievement. *J. Biomechanics* (in press).

Soest, A. J. van and Bogert, A. J. van den (1991) Criteria for the comparison of direct dynamics software systems to be used in the field of biomechanics. In *Proc. 3rd Int. Symp. on Computer Simulation in Biomech.* University of Western Australia, Perth.

Werff, K. van der (1977) Kinematic and dynamic analysis of mechanisms, a finite element approach. Ph.D. thesis, Delft University Press, Delft.

Zajac, F. and Winters, J. M. (1990) Modeling musculoskeletal movement systems: joint and body-segment dynamics, musculotendinous actuation and neuromuscular control. In *Multiple Muscle Systems* (Edited by Winters, J. M. and Woo, S. L.-Y), pp. 121–148. Springer, New York.

APPENDIX
MATHEMATICAL FORMULATION OF SPACAR

NOMENCLATURE

$\mathbf{x}$	Column vector of coordinates of a single element
$\mathbf{e}$	Column vector of deformations of a single element
$\mathbf{X}$	Column vector of coordinates of an entire mechanical system
$\mathbf{E}$	Column vector of deformations of all elements in a system

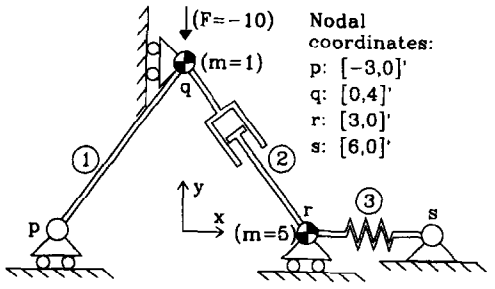

Fig. A1. The planar mechanical system used in the numerical example throughout the Appendix. Encircled numbers indicate element number; force F in Newtons; masses m in kilograms.

$\mathbf{X}_0, \mathbf{X}_c, \mathbf{X}_m$	Fixed, calculable and input parts of the system **X** vector
$\mathbf{E}_0, \mathbf{E}_c, \mathbf{E}_m$	Fixed, calculable and input parts of the system **E** vector
C	Continuity function defining $\mathbf{e}=C(\mathbf{x})$
F	Zero-order transfer function defining $\mathbf{X}=F(\mathbf{X}_m, \mathbf{E}_m)$
D, D^2	First, second differential operator
δ, δ^2	First, second variation
F	Column vector of external forces applied at **X**
S	Column vector of stresses due to deformations **E**
$\dot{\mathbf{X}}, \dot{\mathbf{E}}$	First derivative of **X**, **E** with respect to time
$\ddot{\mathbf{X}}, \ddot{\mathbf{E}}$	Second derivative of **X**, **E** with respect to time

The formulation of the method will be limited to the two-dimensional case. It is emphasized that the method is fully developed for three-dimensional systems as well. The method will be illustrated with a numerical example. As the example presented in the main text is too complicated for this purpose, another example system is presented in Fig. A1. This example does not represent any real physical system; it is merely constructed to illustrate the aspects of SPACAR introduced here.

BASICS

(1) The position of an element is described by a column vector of coordinates **x**.

Example. The position of element 1 (Fig. A1) is given by $[x_p, y_p, x_q, y_q]^T$.

(2) A *type of element* is defined by its number of coordinates and the way in which the column vector of deformations **e** of the element type depends on **x**.

Example. The element type 'TRUSS', of which elements 1, 2, 3 (Fig. A1) are examples, has four coordinates: the x and y coordinates of its end points. Three are needed to define the position of a rigid TRUSS. Therefore, one deformation mode should be defined: elongation along a straight line.

(3) Deformations are a function of the coordinates of the element. This function is called the *continuity equation* of the element type:

$$\mathbf{e}=\mathrm{C}(\mathbf{x}). \tag{A1}$$

These continuity equations are highly nonlinear.

Example. The instantaneous length of the TRUSS 2 with end points q, r is:

$$l_2=\sqrt{(x_r-x_q)^2+(y_r-y_q)^2}.$$

When its rest length l_0 is defined to be 2.0, the continuity equation is:

$$e_2=\sqrt{(x_r-x_q)^2+(y_r-y_q)^2}-l_0 \text{ yielding}$$

$$e_2=\sqrt{9+16}-2.0=3.0.$$

(4) First and second derivatives of the continuity equations of each type of element can be obtained, yielding:

$$\delta\mathbf{e}=\mathrm{DC}(\mathbf{x})\delta\mathbf{x}, \tag{A2}$$

$$\delta^2\mathbf{e}=\mathrm{DC}(\mathbf{x})\,\delta^2\mathbf{x}+\mathrm{D}^2C(\mathbf{x})\delta\mathbf{x}^2, \tag{A3}$$

where DC and D^2C are position-dependent *matrices of partial derivatives* ('Jacobians') defining linear and bilinear maps, respectively.

Example. Differentiating the continuity equation for TRUSS 1 with respect to its nodal point coordinates $\mathbf{x}=[x_p, y_p, x_q, y_q]^T$ yields:

$$\mathrm{DC}(\mathbf{x})=[-\cos(\varphi), -\sin(\varphi), \cos(\varphi), \sin(\varphi)],$$

where $\varphi=\arctan[(y_q-y_p)/(x_q-x_p)]$ is introduced to simplify notation. Thus, in the example position this yields for element 1:

$$\mathrm{DC}([x_p, y_p, x_q, y_q]^T)=[-0.6, -0.8, 0.6, 0.8] \text{ and}$$

$$\delta e_1=[-0.6, -0.8, 0.6, 0.8][\delta x_p, \delta y_p, \delta x_q, \delta y_q]^T.$$

(5) For a multilink system, vectors **X** and **E** (note capitals) can be defined to be the union of all element **x** vectors (coordinates may be shared by elements) and the juxtaposition of all element **e** vectors (deformations are not shared).

(6) Both **X** and **E** can be split into three distinct parts: *fixed* (subscript '0', changes are zero), *calculable* (subscript 'c') and *input* (subscript 'm', the independent movable variables).

Example. The system has two degrees of freedom, chosen to be x_p and e_3; thus,

$$\mathbf{X}=[x_p, y_p, x_q, y_q, x_r, y_r, x_s, y_s]^T, \quad \mathbf{E}=[e_1, e_2, e_3]^T,$$

which can be split into $\mathbf{X}_0, \mathbf{X}_c, \mathbf{X}_m$ and $\mathbf{E}_0, \mathbf{E}_c, \mathbf{E}_m$, respectively:

$$\mathbf{X}_0=[y_p, x_q, y_r, x_s, y_s]^T, \mathbf{X}_c=[y_q, x_r]^T, \mathbf{X}_m=[x_p],$$

$$\mathbf{E}_0=[e_1], \quad \mathbf{E}_c=[e_2], \quad \mathbf{E}_m=[e_3].$$

(7) The input variables $\mathbf{X}_m, \mathbf{E}_m$ by definition solely determine **X** and **E**:

$$\mathbf{X}=F_X(\mathbf{X}_m, \mathbf{E}_m), \quad \mathbf{E}=F_E(\mathbf{X}_m, \mathbf{E}_m). \tag{A4}$$

The unknown maps F_X and F_E are highly nonlinear. They are called *zero-order transfer functions.*

(8) First and second derivatives of the zero-order transfer function must exist: the first-($\mathrm{D}F_X, \mathrm{D}F_E$) and second-($\mathrm{D}^2F_X, \mathrm{D}^2F_E$) order transfer functions, respectively:

$$\delta\mathbf{X}=\mathrm{D}F_X\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T, \tag{A5}$$

$$\delta\mathbf{E}=\mathrm{D}F_E\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T, \tag{A6}$$

$$\delta^2\mathbf{X}=\mathrm{D}F_X\,[\delta^2\mathbf{X}_m^T\,\delta^2\mathbf{E}_m^T]^T + \mathrm{D}^2F_X\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T, \tag{A7}$$

$$\delta^2\mathbf{E}=\mathrm{D}F_E\,[\delta^2\mathbf{X}_m^T\,\delta^2\mathbf{E}_m^T]^T + \mathrm{D}^2F_E\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T\,[\delta\mathbf{X}_m^T\,\delta\mathbf{E}_m^T]^T. \tag{A8}$$

KINEMATICS

The kinematics problem is to calculate the movement of the entire system given the movement of the degrees-of-

freedom $\mathbf{X}_m$, $\mathbf{E}_m$. In other words: to obtain the maps F_X, F_E, DF_X, DF_E, D^2F_X, D^2F_E. To avoid complexity in notation, the second-order transfer function will not be derived.

Combining equations (A1) and (A4), we obtain for the entire system

$$F_E(\mathbf{X}_m, \mathbf{E}_m) = C(F_X(\mathbf{X}_m, \mathbf{E}_m)) \tag{A9}$$

and similarly, combining equations (A2), (A5) and (A6),

$$DF_E(\mathbf{X}_m, \mathbf{E}_m) = DC\,DF_X(\mathbf{X}_m, \mathbf{E}_m). \tag{A10}$$

Using equation (A9), the calculable deformations $\mathbf{E}_c$ (which concern overdetermined parts of the system) can be obtained as soon as F_X and thus $\mathbf{X}$ is known.

Example. In the starting position of the example system, where all coordinates are given, e_2 can be calculated using the continuity equation of this element:

$$e_2 = \sqrt{(x_r - x_q)^2 + (y_r - y_q)^2} - l_{02}.$$

When the part concerning $\mathbf{E}_c$ is removed from equation (A10) and the remaining part is split up in the way introduced before we can write:

$$\begin{bmatrix} DF_{E_0} \\ DF_{E_m} \end{bmatrix} = \begin{bmatrix} D_{X_0}C_{E_0} \\ D_{X_0}C_{E_m} \end{bmatrix} DF_{X_0} + \begin{bmatrix} D_{X_c}C_{E_0} \\ D_{X_c}C_{E_m} \end{bmatrix} DF_{X_c} + \begin{bmatrix} D_{X_m}C_{E_0} \\ D_{X_m}C_{E_m} \end{bmatrix} DF_{X_m}, \tag{A11}$$

where, for example, $D_{X_0}C_{E_m}$ stands for the matrix of partial derivatives $[\partial \mathbf{E}_m / \partial \mathbf{X}_0]$.

When $\mathbf{X}$ is entirely known (which is supposed to be the case in the starting position) all partial DC matrices can be calculated.

Example. Using the first derivative of the continuity equation for TRUSS derived earlier ($DC(\mathbf{x}) = [-\cos(\varphi), -\sin(\varphi), \cos(\varphi), \sin(\varphi)]$), the partial DC matrices in the present position are:

$$\begin{aligned}
D_{X_0}C_{E_0} &= D_{y_p, x_q, y_r, x_s, y_s}C_{e_1} = [-0.8,\ 0.6,\ 0,\ 0,\ 0], \\
D_{X_0}C_{E_m} &= D_{y_p, x_q, y_r, x_s, y_s}C_{e_3} = [0,\ 0,\ 0.0,\ 1.0,\ 0.0], \\
D_{X_c}C_{E_0} &= D_{y_q, x_r}C_{e_1} = [0.8,\ 0], \\
D_{X_c}C_{E_m} &= D_{y_q, x_r}C_{e_3} = [0,\ -1.0], \\
D_{X_m}C_{E_0} &= D_{x_p}C_{e_1} = [-0.6], \\
D_{X_m}C_{E_m} &= D_{x_p}C_{e_3} = [0].
\end{aligned}$$

Furthermore, the following partial DF matrices are known by definition:

$$\begin{aligned}
DF_{E_0} &= [\partial \mathbf{E}_0 / \partial \mathbf{X}_m\ \partial \mathbf{E}_0 / \partial \mathbf{E}_m] = [\{0\}, \{0\}], \\
DF_{E_m} &= [\partial \mathbf{E}_m / \partial \mathbf{X}_m\ \partial \mathbf{E}_m / \partial \mathbf{E}_m] = [\{0\}, \{1\}], \\
DF_{X_0} &= [\partial \mathbf{X}_0 / \partial \mathbf{X}_m\ \partial \mathbf{X}_0 / \partial \mathbf{E}_m] = [\{0\}, \{0\}], \\
DF_{X_m} &= [\partial \mathbf{X}_m / \partial \mathbf{X}_m\ \partial \mathbf{X}_m / \partial \mathbf{E}_m] = [\{1\}, \{0\}],
\end{aligned}$$

where, for example $[\{0\}, \{1\}]$ stands for a matrix consisting of blocks of zeros and ones, respectively. Thus the only unknown in equation (A11) is the partial map DF_{X_c}. Solving for DF_{X_c} yields:

$$DF_{X_c} = \begin{bmatrix} D_{X_c}C_{E_0} \\ D_{X_c}C_{E_m} \end{bmatrix}^{-1} \left[\begin{bmatrix} DF_{E_0} \\ DF_{E_m} \end{bmatrix} - \begin{bmatrix} D_{X_m}C_{E_0} \\ D_{X_m}C_{E_m} \end{bmatrix} DF_{X_m} \right]. \tag{A12}$$

Example. In the position shown, equation (A12) yields:

$$DF_{y_q, x_r} = \begin{bmatrix} 0.8 & 0 \\ 0 & -1.0 \end{bmatrix}^{-1} \left[\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -0.6 \\ 0 \end{bmatrix} [1 \quad 0] \right] = \begin{bmatrix} 0.75 & 0.0 \\ 0.0 & -1.0 \end{bmatrix},$$

$$\begin{bmatrix} \delta y_q \\ \delta x_r \end{bmatrix} = \begin{bmatrix} 0.75 & 0.0 \\ 0.0 & -1.0 \end{bmatrix} \begin{bmatrix} \delta x_p \\ \delta e_3 \end{bmatrix}.$$

The first-order transfer function has been determined; the second-order transfer function can be determined in a similar way. Unfortunately, the zero-order transfer function cannot be obtained in a similar way. However, given an initial position $\mathbf{X}_{old}$ and the desired change in $\mathbf{X}_m$, $\mathbf{E}_m$, a new position $\mathbf{X}_{new}$ can be estimated after calculation of DF_x and D^2F_x using a truncated Taylor expansion:

$$\begin{aligned}
\mathbf{X}_{new} &\approx \mathbf{X}_{old} + DF_X\,[\delta \mathbf{X}_m^T\ \delta \mathbf{E}_m^T]^T \\
&\quad + D^2F_X\,[\delta \mathbf{X}_m^T\ \delta \mathbf{E}_m^T]^T\,[\delta \mathbf{X}_m^T\ \delta \mathbf{E}_m^T]^T, \\
\mathbf{E}_{new} &= C(\mathbf{X}_{new}).
\end{aligned} \tag{A13}$$

The error made in this approximation is reflected in the amount by which the $\mathbf{E}_0$ part of $\mathbf{E}_{new}$ differs from zero. An iteration scheme is used to reduce this error. In most cases convergence to a relative error smaller than 0.00001 occurs within five iterations.

Example. Suppose we wish to know $\mathbf{X}$ for $[\mathbf{X}_m^T\ \mathbf{E}_m^T]^T = [x_p\ e_3]^T = [-2.9\ 1.1]^T$. As $[\mathbf{X}_m^T\ \mathbf{E}_m^T]^T$ in the known starting position is $[-3, 1]^T$ if we define the rest length of element 3 to be 2, the desired $[\delta \mathbf{X}_m^T\ \delta \mathbf{E}_m^T]^T = [\delta x_p\ \delta e_3]^T = [0.1, 0.1]^T$. Using the already calculated DF_{X_c} we can estimate $X_{c_{new}}$ (omitting the second-order term) to be:

$$\begin{bmatrix} y_{q_{new}} \\ x_{r_{new}} \end{bmatrix} \approx \begin{bmatrix} 4.0 \\ 3.0 \end{bmatrix} + \begin{bmatrix} 0.75 & 0.0 \\ 0.0 & -1.0 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 4.075 \\ 2.9 \end{bmatrix}.$$

Using the TRUSS continuity equation, e_1 (which should be zero) is calculated to be $\sqrt{25.015625} - 5 \approx 0.00156$.

DYNAMICS

The second part of the problem concerns the derivation of the equations of motion governing the behavior of the system.

In the derivation of the equations of motion two concepts are combined:

—the *first- and second-order transfer functions* as described earlier, which together determine the velocity and acceleration of the entire system as a function of the velocity and acceleration of the degrees-of-freedom;

—the *principle of virtual work* which in its most basic form states (Meriam, 1975) that "The virtual work done by external active forces on an ideal mechanical system in equilibrium is zero for any and all virtual displacements." Extending this principle to include elastically deformable elements and inertial forces, it can be written in matrix form as:

$$\delta \mathbf{X}^T [\mathbf{F} + \mathbf{F}_{inertia}] = \delta \mathbf{E}^T \mathbf{S}, \tag{A14}$$

which should hold for all kinematically allowed $\delta \mathbf{X}$, $\delta \mathbf{E}$; $\mathbf{F}$, $\mathbf{S}$ and $\mathbf{F}_{inertia}$ represent the vectors of active forces, internal stresses and inertial forces, respectively.

The general term $\mathbf{F}_{inertia}$ can be replaced by $-\mathbf{M}\ddot{\mathbf{X}}$ (with $\mathbf{M}$ a mass matrix) since this is the only inertial term.

Example. With point masses of 1 and 5 kg at nodes q and r, respectively, the mass matrix is given by: diag

({0, 0, 1, 1, 5, 5, 0, 0}), where diag (**a**) is defined to be a square matrix with nonzero entries equalling **a** along the main diagonal.

Combining equation (A14) with equations (A5) and (A6) yields:

$$[\delta \mathbf{X}_M^T \delta \mathbf{E}_M^T]\, DF_X^T [\mathbf{F} - \mathbf{M}\ddot{\mathbf{X}}] = [\delta \mathbf{X}_m^T \delta \mathbf{E}_m^T]\, DF_E^T \mathbf{S},$$

which should hold for all kinematically allowed $[\delta \mathbf{X}_M^T \delta \mathbf{E}_m^T]$. However, this vector concerns the degrees-of-freedom only, and may, therefore, by definition have any value. Therefore,

$$DF_X^T [\mathbf{F} - \mathbf{M}\ddot{\mathbf{X}}] = DF_E^T \mathbf{S}. \tag{A15}$$

By combining equations (A15) and (A7) we obtain after some rearrangement:

$$[DF_X^T \mathbf{M} DF_X] \cdot \begin{bmatrix} \ddot{\mathbf{X}}_m \\ \ddot{\mathbf{E}}_m \end{bmatrix} = [DF_X^T \mathbf{F}] - [DF_E^T \mathbf{S}] - DF_X^T \mathbf{M} D^2 F_X \begin{bmatrix} \dot{\mathbf{X}}_m \\ \dot{\mathbf{E}}_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{X}}_m \\ \dot{\mathbf{E}}_m \end{bmatrix}.$$

It can be seen that singularity of the part $[DF_X^T \mathbf{M} DF_X]$ occurs when modalities of movement are present that have no mass attached to them.

Under the condition that $[DF_X^T \mathbf{M} DF_X]$ is nonsingular this equation can be solved for $\ddot{\mathbf{X}}_m$, $\ddot{\mathbf{E}}_m$:

$$\begin{bmatrix} \ddot{\mathbf{X}}_m \\ \ddot{\mathbf{E}}_m \end{bmatrix} = [DF_X^T \mathbf{M} DF_X]^{-1} \left[[DF_X^T \mathbf{F}] - [DF_E^T \mathbf{S}] - DF_X^T \mathbf{M} D^2 F_X \begin{bmatrix} \dot{\mathbf{X}}_m \\ \dot{\mathbf{E}}_m \end{bmatrix} \begin{bmatrix} \dot{\mathbf{X}}_m \\ \dot{\mathbf{E}}_m \end{bmatrix} \right]. \tag{A16}$$

Example. Suppose an external force $F_{y_q} = -10$ N is present. Further, let's suppose that the stresses in elements 2, 3 are governed by $S_2 = 100 \dot{e}_2$; $S_3 = 50 e_3$. Finally, let's suppose the velocity of the system is zero. Now we set up equation (A16), omitting the parts concerning $\mathbf{X}_0$, $\mathbf{E}_0$.

$$\mathbf{F}_{y_q, x_r, x_p} = [-10, 0, 0]^T, \quad \mathbf{M}_{y_q, x_r, x_p} = \text{diag}(\{1, 5, 0\}),$$
$$\mathbf{S}_{e_2, e_3} = [0, 50]^T,$$

$$DF_{y_q, x_r, x_p} = \begin{bmatrix} DF_{X_c} \\ DF_{X_m} \end{bmatrix} = \begin{bmatrix} 0.75 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix},$$

$$DF_{e_2, e_3} = \begin{bmatrix} DF_{E_c} \\ DF_{E_m} \end{bmatrix} = \begin{bmatrix} 0.6 & -0.6 \\ 0 & 1 \end{bmatrix}.$$

Because of the zero velocity, the last part of equation (A16) can be omitted:

$$\begin{bmatrix} \ddot{x}_p \\ \ddot{e}_3 \end{bmatrix} = \left[\begin{bmatrix} 0.75 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.75 & 0 \\ 0 & -1 \\ 1 & 0 \end{bmatrix} \right]^{-1} \left[\begin{bmatrix} 0.75 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -10 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.6 & 0 \\ -0.6 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 50 \end{bmatrix} \right],$$

$$\begin{bmatrix} \ddot{x}_p \\ \ddot{e}_3 \end{bmatrix} = \begin{bmatrix} 0.5625 & 0 \\ 0 & 5 \end{bmatrix}^{-1} \left[\begin{bmatrix} -7.5 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 50 \end{bmatrix} \right] = \begin{bmatrix} 1.778 & 0 \\ 0 & 0.2 \end{bmatrix} \begin{bmatrix} -7.5 \\ -50 \end{bmatrix} = \begin{bmatrix} -13.33 \\ -10.00 \end{bmatrix},$$

which can by inspection be seen to be correct. Note from this example, that (as expected) the second-order transfer function determines the velocity dependency of the accelerations.

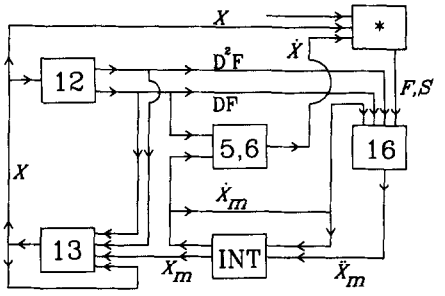

Fig. A2. Block diagram of calculations in SPACAR. Numbers in blocks refer to equation numbers in the appendix. The block marked INT represents the integration algorithm; the block marked * may contain any (user-defined) functions defining forces and stresses.

SUMMARY

The entire calculation process can be summarized as follows (Fig. A2). Given a position of the entire mechanism at time $= t$, first- and second-order transfer functions are calculated using equation (A12). Using the first-order transfer function, the velocity of the entire system is calculated from the velocity of the degrees of freedom, using equations (A5) and (A6). Now all external forces, internal stresses and driving forces/stresses can be calculated. Next, accelerations of the degrees-of-freedom are calculated using equation (A16). The accelerations and velocities of the defrees-of-freedom are integrated numerically to yield the velocity and position of the degrees-of-freedom at time $= t + \delta t$. Finally, the new position of the entire mechanism is calculated using equation (A13).

At present, a variable-order variable-stepsize Adams–Bashford predictor/Adams–Moulton corrector integration algorithm (Shampine and Gordon, 1975) is used for numerical integration. This algorithm is efficient and robust; furthermore, it has been shown here that it yields correct solutions for unstable systems. Nevertheless, it can be substituted by any other integration algorithm if desired.

Although this may not be obvious from the example used, it must be realized that the vectors **F** and **S** can be any (user-defined) function of time, position and velocity of the mechanical system. Also, actuators (for example, torque actuators operating on a joint angle) can be incorporated using standard elements. An efficient way to incorporate Hill-type muscle models as actuators has been developed [see van Soest *et al.* (1992)].

It is obvious that the potential of a method like SPACAR depends strongly on the element types available. At present, the set of element types allows description of a large variety of mechanical systems. However, in case 'special-purpose elements' are needed, they can be incorporated in SPACAR if their continuity equations plus first and second derivatives are available. Examples of development and use of highly complex special purpose elements in kinematical analysis are given by van der Helm (1991).