Multibody System Order n Dynamics Formulation Based on Velocity Transform Method

J. E. Keat*
Cambridge Research, A Division of Photon Research Associates, Cambridge, Massachusetts

In the multibody-simulation discipline, equations-of-motion formulations in which the number of calculations per integration step increase only linearly with the number n of bodies are called order n or $\mathcal{O}(n)$ formulations. The development of such formulations is an area of current research because of their capability of yielding simulations that run much faster than conventional ones when n is large. This paper presents a new order n algorithm. It is applicable to systems of rigid or nonrigid bodies. It permits the bodies' interconnection joints to have an arbitrary number of degrees of freedom between 0 and 6. The system can have open-chain, tree, or closed-loop topology. Both absolute accelerations and relative accelerations are established by the technique. Closed topological loops are handled by the concept of cut joints. Constraint forces are calculated only at the cut joints, not at the uncut ones. The derivation of the algorithm uses a velocity transformation to eliminate the appearance of forces due to constraints at the uncut joints. The algorithm entails sequential computational passes—backward and forward—through the system of bodies.

Introduction

COMPUTER programs that simulate the dynamics of multibody systems are noted for slow running speed, particularly when the number n of bodies is large. The difficulty results, in part, from the way in which the system's kinetics equations are formulated. Most multibody programs employ a single large kinetics equation of the form

$$M'\dot{v}' = f' \tag{1}$$

The primes are employed in Eq. (1) for maximal compatibility with notation used subsequently. Overdot denotes differentiation with respect to time; $\dot{v}'$ is composed of subvectors $\dot{v}_i$, $i = 1$ to n pertaining to the velocities of the individual bodies Z_i of the system. The generalized mass matrix is M'. The f' contains the external, internal, and dynamic forces acting on the system. Equation (1) must be solved for $\dot{v}'$ one or more times in each integration step, the number being dependent on the numerical integration algorithm. The number of operations required to solve Eq. (1) is a third-degree polynomial in n, if a direct method of solution is used and all v_i' have the same dimension. The number of operations needed to compute M' usually is a third-degree polynomial in n. Hence, the total number of operations per integration step is a third-degree polynomial in n. Programs employing Eq. (1) are therefore said to be $\mathcal{O}(n^3)$. For n to be sufficiently large, forming and solving Eq. (1) will be the most time-consuming operation performed by the program in each integration step and hence the dominant cause of slow running speed.

An important area of current research in multibody dynamics is concerned with $\mathcal{O}(n)$ algorithms. The number of operations per integration step needed by programs using an $\mathcal{O}(n)$ algorithm is only a first degree polynomial in n, when applied to multibody systems in which all v_i' vectors have the same dimension. Hence, for n sufficiently large, an $\mathcal{O}(n)$ program will run much faster than conventional current multibody programs. For example, in a study by Sherman[1] an $\mathcal{O}(n)$ algorithm reduced the number of calculations per integration step by a factor of 4 when applied to a multibody system with 15 degrees of freedom (DOF).

With $\mathcal{O}(n)$ algorithms, the vector elements $\dot{v}_i'$ of $\dot{v}'$ are not calculated simultaneously via a single kinetics equation of the form of Eq. (1). Instead, they are computed sequentially. The operation requires two or more body-by-body passes through the system of bodies. Backward passes as well as forward passes are employed. The $\dot{v}_i'$ are computed, body-by-body, in the final pass. The purpose of the earlier passes is to calculate the values of terms needed in the final pass.

Work on $\mathcal{O}(n)$ algorithms prior to 1986 dealt mainly with robot arms. A short bibliography of these studies has been presented by Rodriguez.[2] His work has been devoted to developing $\mathcal{O}(n)$ algorithms that bear a mathematical similarity to equations used in estimation and control theory. His initial work[2] was limited to systems of rigid bodies connected by pin joints in an open-chain topology with one end-body free and the other pin jointed to a fixed base. His later studies[3-5] delve into systems with tree and closed-loop topologies.

A computer program, SDFAST, which employs symbolic manipulation to establish tailor-made $\mathcal{O}(n)$ equations for individual systems, has been described by Sherman.[1] Rosenthal[6] discusses its application to a simple 2 DOF planar pendulum. The program is restricted to rigid bodies and to systems with less than 51 DOF. It is applicable to joints with 1 DOF in translation or 1, 2, or 3 DOF in rotation. Open-chain, tree, and closed-loop topologies can be handled by the program.

The work of Haug[7] on $\mathcal{O}(n)$ algorithms was motivated by the desire to establish a technique that is suitable for implementation on parallel processors. $\mathcal{O}(n)$ algorithms have the potential for being implemented efficiently on parallel processors when the multibody system has tree topology, since calculations pertaining to individual branches of the tree can, in part, be performed in parallel rather than sequentially. Haug's formulation permits individual bodies to be rigid or nonrigid. The joints can have 1 DOF in translation and/or 1 DOF in rotation. The interconnection topology can be open chain, tree, or closed loop. Haug derived the formulation via D'Alembert's principle of virtual work.

Received March 17, 1988; revision received Sept. 1, 1988.

*Staff Scientist. Member AIAA.

Schwertassek[8] presents an $\mathcal{O}(n)$ formulation for an open chain of rigid bodies connected by pin joints. His paper includes bibliographic material on $\mathcal{O}(n)$ work done in Europe.

The present paper presents a new $\mathcal{O}(n)$ algorithm. The algorithm uses basic multibody system modeling concepts employed previously by the author,[9-13] but the mathematics is altered radically in order to compute the vector elements $\dot{v}_i'$ of $\dot{v}'$ sequentially. The development uses a modification of Jerkovsky's idea[14] of applying a velocity transformation to the equations of motion of a "primitive" system. Closed topological loops are handled through the concept of joint cuts. Interbody constraint forces are computed only at joints that are cut; the formulation differs from that of Rodriguez in this respect, since Rodriguez' method requires calculating the constraint forces at all joints. In addition to calculating the acceleration vectors $\dot{v}_i'$ that are to be integrated numerically to obtain velocity vectors v_i', the present approach also computes a second set of acceleration vectors $\dot{v}_i$. The $\dot{v}_i$ are auxiliary vectors that are introduced only because they are needed to calculate the $\dot{v}_i'$. The distinction between the $\dot{v}_i'$ and the $\dot{v}_i$ will be made apparent subsequently in the paper.

The $\mathcal{O}(n)$ algorithm described in this paper permits the individual bodies to be rigid or nonrigid. The joints can have an arbitrary number of DOF between 0 and 6. The topology can be open chain, tree, or closed loop. The system can be free in space or attached to a fixed base through joints. Of the $\mathcal{O}(n)$ algorithms developed previously, only Haug's[7] approaches the present one in generality. The present formulation differs significantly from Haug's in regard to the variables and equations that are used during the sequential passes and in the manner of computing the Lagrange multiplier vector λ, which is employed to determine the constraint forces at the cut joints. Also, the present algorithm is presented in more detail than that of Haug in Ref. 7.

Topology

Consider a generic system of n bodies that are interconnected by joints as indicated on Fig. 1. Let this system be designated by the symbol Σ. The bodies can be rigid or nonrigid, and the joints can have from 0 to 6 DOF. Introduce a graph G to indicate the topology of Σ. In drawing G, the bodies are the nodes, and the joints are the links. Introduce an inertial reference frame, which will be designated as frame N. Two conditions will be considered: 1) Σ free in space and 2) Σ restrained through joints between frame N and one or more bodies. If Σ is free in space, consider a selected body to be connected to frame N through a 6-DOF joint. N and all joints connected to N are to be included on G.

If G has closed loops, introduce the minimum number of joint cuts needed to eliminate them. Let this new system of bodies, uncut joints, and cut joints be denoted by the symbol Σ'. Σ' will contain n bodies and uncut joints. Let m denote the number of its cut joints. Let the symbols Z, B, and Γ, respectively, denote bodies, uncut joints, and cut joints. Attach subscript labels i, j, or k to Z and B to indicate individual bodies and uncut joints. Attach subscript labels α or γ to Γ to denote individual cut joints. i, j, and k have the range 1 to n, and α and γ have the range 1 to m.

Let G' be the graph produced by deleting the Γ from G. This new graph G' will have either open chain or tree topology. It will include one joint between N and a body. Designate this joint as B_1 and the body as Z_1. Number the remaining Z by paths according to the scheme illustrated in Fig. 2. The input and output ends of each B and Γ are defined to be the ends attached to the lowest and highest number Z, respectively. Let each B be given the number of the Z at its output end. The Γ can be numbered arbitrarily.

If, when passing from Z_1 to some body Z_i on G' one goes through Z_j, then Z_j is said to be "inboard" of Z_i, and Z_i is said to be "outboard" of Z_j. Let S_{Ii} and S_{Oi} be integer sets containing 1) the numbers of all bodies inboard and outboard of Z_i, respectively, and 2) the number i itself. Employ the symbol $i-1$ to designate the number of the body immediately inboard of Z_i. It is evident from Fig. 2 that $i-1$ is not always precisely one less than i if G' is a tree rather than merely an open chain. Let $i+1(k)$ designate the number of the body immediately outboard of Z_i on the path to Z_k, where $k \in S_{Oi}$ and $k \neq i$. Let S_{i+1} be the integer set containing the numbers of all bodies immediately outboard of Z_i. S_{i+1} is the null set if Z_i is the terminal body of a path. If G' is an open chain, S_{i+1} will contain precisely one element for all Z_i except the terminal body Z_n.

Kinematics

Specify a body reference frame ζ_i for each Z_i. The ζ_i is attached to Z_i in a prescribed manner. The velocity state of Z_i relative to frame N can be specified by the following column vector v_i:

$$v_i = \begin{Bmatrix} v_{\zeta_i} \\ w_{\zeta_i} \\ \dot{\eta}_i \end{Bmatrix} \tag{2}$$

where v_{ζ_i} and ω_{ζ_i} are 3×1 column vectors of the france ζ_i components of the translational and angular velocity Gibbs vectors, respectively, of Z_i relative to N. The η_i is a column vector of generalized coordinates that define the time varying mass distribution of Z_i relative to ζ_i. The elements of η_i com-

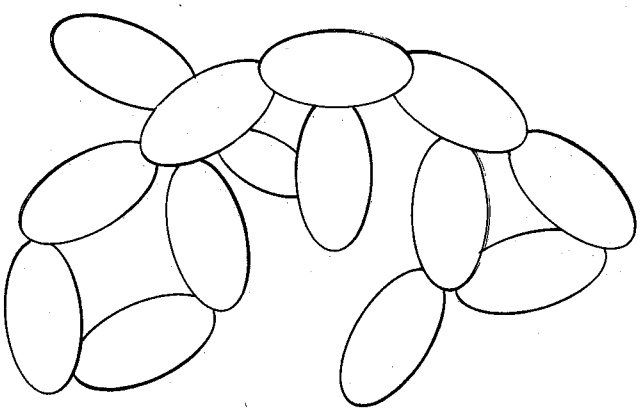

Fig. 1 Sample multibody system.

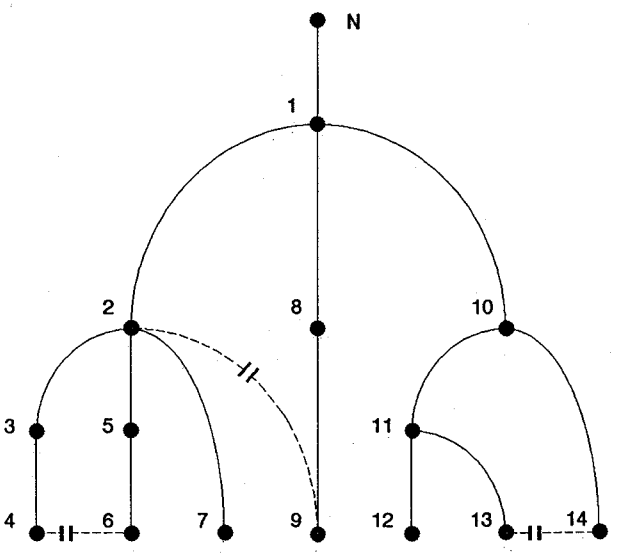

Fig. 2 Sample graph G' to illustrate body numbering scheme.

monly are selected to be the amplitudes of normal vibration modes of Z_i. Let the composite column vector that is produced by stacking the v_i in numerical order be designated as v.

Let β_{jI} and β_{jO} be a pair of coordinate frames at the input and output ends of B_j. These frames are attached to Z_{j-1} and Z_j, respectively. Let p_j and ρ_j be column vectors defining the configuration and velocity, respectively, of β_{jO} relative to β_{jI}. The dimension of ρ_j is equal to the number of DOF of B_j. Let the velocity condition of Z_j relative to β_{jI} be specified by the composite column vector v_j'

$$v_j' = \begin{Bmatrix} \rho_j \\ \dot{\eta}_j \end{Bmatrix} \tag{3}$$

Let the composite vector that is produced by stacking the v_j' in numerical order be designated as v'.

The v_i are related to the v_j' by

$$v_i = \sum_{j \in S_{Ii}} T_{ij} v_j' \tag{4}$$

where $i = 1$ to n and S_{Ii} is an integer set that was defined in the preceding section. The T_{ij} is a matrix that is a function of the (p_k, η_k) of all $k \in S_{Ii} \cap S_{Ij}$. Stacking Eq. (4) for all Z_i yields

$$v = Tv' \tag{5}$$

where v and v' are column vectors that were defined previously. The T is a matrix composed of the submatrices T_{ij}. Our body-numbering scheme is such that T is block lower triangular in the T_{ij}.

The v_i are related to the v_{i-1} and v_i' by

$$v_i = R_{i,i-1} v_{i-1} + T_{ii} v_i' \tag{6}$$

where the velocity transformation matrix $R_{i,i-1}$ is a function of η_{i-1}, p_i, and η_i. Differentiating Eq. (6) with respect to time produces

$$\dot{v}_i = R_{i,i-1} \dot{v}_{i-1} + T_{ii} \dot{v}_i' + b_i \tag{7}$$

with

$$b_i = \dot{R}_{i,i-1} v_{i-1} + \dot{T}_{ii} v_i' \tag{8}$$

Equation (7) is an important kinematics equation that will be used subsequently.

For $i \in S_{Ij}$, define R_{ji} to be

$$R_{ji} = R_{j,j-1} R_{j-1,j-2} \cdots R_{i+1(j),i} \tag{9}$$

and define $R_{ii} = I$, where I is the unit matrix. It can be shown that

$$T_{ji} = R_{ji} T_{ii} \tag{10}$$

Using the T_{ji} to form T and introducing Eq. (10) yields

$$T = RT_D \tag{11}$$

where T_D is a block diagonal matrix whose diagonal blocks are the T_{ii}. The R is a block lower triangular with blocks R_{ji}. The matrices R and T have the same structure in regard to blocks T_{ji} and R_{ji}, which are zero. That is, $R_{ji} = 0$ and $T_{ji} = 0$ for $i \notin S_{Ij}$.

Constraints

Let j_{α_I} and j_{α_O} be the numbers of the bodies at the input and output ends of cut joint Γ_α. If Γ_α is constrained, it will have between 1 and 6 scalar constraints on the motion of its output frame α_O relative to its input frame α_I. These will be modeled as a vector c_α of velocity constraints

$$c_\alpha = C_{\alpha j_{\alpha_I}} v_{j_{\alpha_I}} + C_{\alpha j_{\alpha_O}} v_{j_{\alpha_O}} = 0 \tag{12}$$

The first of the two terms in the middle of Eq. (12) is to be omitted if the input end of Γ_α is connected to frame N rather than to a body. Equation (12) can be written in terms of v as

$$c_\alpha = C_\alpha v = 0 \tag{13}$$

Let c^Γ be the composite column vector that is produced by stacking Eq. (13) for all cut joints which have constraints. Hence,

$$c^\Gamma = C^\Gamma v = 0 \tag{14}$$

Constraints at uncut joints will be modeled in the same manner as constraints at cut joints. Proceeding as before yields the following equation for the constraints at the uncut joints

$$c^B = C^B v = 0 \tag{15}$$

where c^B is a column vector comprising all constraints of all uncut joints, and C^B is the corresponding constraint matrix.

Kinetics

The kinetics equations for Z_i can be written in compact form

$$M_i \dot{v}_i = f_i^B + f_i^\Gamma + f_i \tag{16}$$

where M_i is the symmetric generalized mass matrix of Z_i. The f_i^B and f_i^Γ are the generalized force vectors on Z_i due to the constraints at the uncut and cut joints, respectively, which are attached to Z_i. The remaining generalized forces on Z_i—external, internal, and inertial—have been combined into the single vector f_i. Equation (16) is derived and discussed in Ref. 11.

Stacking Eqs. (16) for the n bodies Z_i and expressing the result in a compact form yields

$$M\dot{v} = f^B + f^\Gamma + f \tag{17}$$

where M is symmetric and block diagonal in the M_i.

Premultiply Eq. (17) by the transpose T^T of the velocity transformation matrix T that was defined earlier. With the constraints at uncut joints defined as in Eq. (15), it can be shown[9] that $T^T f^B = 0$. Hence, the constraint forces at the uncut joints drop out of the formulation. Introducing Eq. (11) then yields

$$T_D^T R^T M \dot{v} = T_D^T f^* + T_D^T R^T f^\Gamma \tag{18}$$

where

$$f^* = R^T f \tag{19}$$

The vector element f_i^* of f^* is the generalized force vector on Z_i due to the nonconstraint forces f_k for all $k \in S_{Oi}$. Because R is block lower triangular with $R_{ii} = I$ enables Eq. (19) to be manipulated into

$$f_i^* = f_i + \sum_{j \in S_{i+1}} R_{ji}^T f_j^* \tag{20}$$

for $i = 1$ to n. Equation (20) is convenient for computational purposes, since it enables the vector elements f_i^* of f^* to be calculated (backward) sequentially.

The previously noted structure of the matrices T_D, R, and M enables Eq. (18) to be written as

$$T_{ii}^T \sum_{j \in S_{Oi}} R_{ji}^T M_j \dot{v}_j = T_{ii}^T f_i^* + T_{ii}^T \sum_{j \in S_{Oi}} R_{ji}^T f_j^\Gamma \tag{21}$$

for $i = 1$ to n. This form is convenient for developing the $\mathcal{O}(n)$ algorithm as discussed in the next section.

Sequential Equations for the Acceleration Vectors

The remaining problem is to manipulate Eqs. (7) and (21) into a form that enables $\dot{v}_i'$ to be computed sequentially. For

convenience in the derivation, we temporarily make the restriction that G' is merely an open chain, rather than a tree. Equation (21) then can be written in the form

$$T_{ii}^T \sum_{j=i}^{n} R_{ji}^T M_j \dot{v}_j = T_{ii}^T f_i^* + T_{ii}^T \sum_{j=i}^{n} R_{ji}^T f_j^\Gamma \tag{22}$$

for $i=1$ to n.

Start with the final, $i=n$, equation of Eqs. (7) and (22), i.e.,

$$\dot{v}_n = R_{n,n-1}\dot{v}_{n-1} + T_{nn}\dot{v}_n' + b_n \tag{23}$$

$$T_{nn}^T R_{nn}^T M_n \dot{v}_n = T_{nn}^T f_n^* + T_{nn}^T R_{nn}^T f_n^\Gamma \tag{24}$$

Substitute Eq. (23) into Eq. (24) and solve for $\dot{v}_n'$ as a function of $\dot{v}_{n-1}$. Then substitute this result into Eq. (23) to obtain an equation for $\dot{v}_n$ as a function of $\dot{v}_{n-1}$. Lastly, substitute this result into Eq. (22) and discard the final, $i=n$, equation; the resulting reduced set of kinetic equations will not include $\dot{v}_n$.

The operations noted in the preceding paragraph are repeated next for the case $i=n-1$. This produces equations for $\dot{v}_{n-1}'$ and $\dot{v}_{n-1}$, as functions of $\dot{v}_{n-2}$, and the reduced set of kinetics equations does not include $\dot{v}_n$ or $\dot{v}_{n-1}$. These operations then are repeated for $i=n-2$, thereby generating equations for $\dot{v}_{n-2}'$ and $\dot{v}_{n-2}$ as a function of $\dot{v}_{n-3}$. At this point, the pattern of the results becomes clear, and one is able to write down the equations for $\dot{v}_i'$ and $\dot{v}_i$ as a function of $\dot{v}_{i-1}$, where i is arbitrary between 1 and n. Lastly, careful inspection of the result enables one to generalize it from open-chain systems to tree systems. The results are as follows:

$$\dot{v}_i' = A_{i,i-1}'\dot{v}_{i-1} + a_i' + N_{ii}' f_i^{\Gamma *} \tag{25}$$

$$\dot{v}_i = A_{i,i-1}\dot{v}_{i-1} + a_i + N_{ii} f_i^{\Gamma *} \tag{26}$$

where

$$A_{i,i-1}' = -N_{ii}' M_i^* R_{i,i-1} \tag{27}$$

$$a_i' = N_{ii}' \left\{ f_i^* - M_i^* b_i - \sum_{j \in S_{i+1}} R_{ji}^T g_j^* \right\} \tag{28}$$

$$N_{ii}' = M_i'^{-1} T_{ii}^T \tag{29}$$

$$f_i^{\Gamma *} = \sum_{j \in S_{Oi}} A_{ji}^T f_j^\Gamma \tag{30}$$

$$A_{i,i-1} = R_{i,i-1} + T_{ii} A_{i,i-1}' \tag{31}$$

$$a_i = b_i + T_{ii} a_i' \tag{32}$$

$$N_{ii} = T_{ii} N_{ii}' \tag{33}$$

$$M_i^* = M_i + \sum_{j \in S_{i+1}} R_{ji}^T M_j^* A_{ji} \tag{34}$$

$$g_j^* = M_j^* a_j + \sum_{k \in S_{j+1}} R_{kj}^T g_k^* \tag{35}$$

$$M_i' = T_{ii}^T M_i^* T_{ii} \tag{36}$$

The superscript asterisk label in the preceding equations denotes terms that are computed sequentially. The M_i^* and M_i' are generalized mass matrices, pertaining to v_i and v_i', respectively, due to the mass of the all bodies Z_j, $j \in S_{Oi}$. Introducing Eq. (31) into Eq. (34) and employing Eqs. (27), (29), and (36) shows that M_i^* as specified by Eq. (34) has the desirable property $M_i^* = M_i^{*T}$. Equations (19) and (33) show that N_{ii} is the transformation, to v_i variables, of the inverse generalized mass matrix $M_i'^{-1}$. In lieu of inverting a large set of linear algebraic equations [Eq. (1)] for the composite system, the algorithm involves the much less time-consuming task of inverting the generalized mass matrix M_i' of each of the n Z_i. The vector g_j^* is the generalized inertial force on Z_j due to the acceleration $\dot{v}_k$ of all Z_k outboard of Z_j.

In the preceding equations, $f_i^{\Gamma *}$ is the generalized force vector on Z_i generated by the constraint force vectors f_j^Γ for all $j \in S_{Oi}$. Equation (30) can be manipulated to produce

$$f_i^{\Gamma *} = f_i^\Gamma + \sum_{j \in S_{i+1}} A_{ji}^T f_j^{\Gamma *} \tag{37}$$

which can be used to compute the $f_i^{\Gamma *}$ (backwards) sequentially. The topic of establishing the f_i^Γ is addressed in the next section.

Computation of the Constraint Force Vectors

This section develops equations for computing the cut-joint constraint force vectors f_i^Γ. Stacking Eqs. (26) for all $i=1$ to n yields

$$\dot{v} = A_S \dot{v} + a + N_D A^T f^\Gamma \tag{38}$$

where N_D is a symmetric block diagonal matrix whose diagonal blocks are the N_{ii}. The first row block of A_S is 0; the remaining row blocks $i=2$ to n each contain precisely one nonzero block: namely $A_{i,i-1}$. The matrix A is block lower triangular with $A_{ii}=I$ and $A_{ji}=0$ for $i \notin S_{Ij}$. For $i \in S_{Ij}$, the A_{ji} have the basic "product" structure indicated previously for R_{ji} in Eq. (9).

It can be shown that

$$(I - A_S)^{-1} = A \tag{39}$$

Hence, solving Eq. (38) for $\dot{v}$ yields

$$\dot{v} = \dot{w} + A N_D A^T f^\Gamma \tag{40}$$

where

$$\dot{w} = Aa \tag{41}$$

The structure of A enables Eq. (41) to be manipulated into

$$\dot{w}_i = A_{i,i-1}\dot{w}_{i-1} + a_i \tag{42}$$

with $i=1$ to n and $\dot{w}_O = \mathbf{0}$. Equation (42) enables the vector elements $\dot{w}_i$ of $\dot{w}$ to be computed sequentially.

It is known from basic principles that f^Γ can be represented as

$$f^\Gamma = C^{\Gamma T} \lambda \tag{43}$$

where λ is a vector of Lagrange multipliers. Differentiating Eq. (15) with respect to time and introducing Eqs. (40) and (43) yields

$$K\lambda = d \tag{44}$$

where

$$d = -\dot{C}^\Gamma v - C^\Gamma \dot{w} \tag{45}$$

$$K = H N_D H^T \tag{46}$$

with

$$H = C^\Gamma A \tag{47}$$

Since $N_D = N_D^T$, it follows that $K = K^T$. When K is nonsingular, Eq. (44) can be solved uniquely for λ, and this result can be used in Eq. (43) to establish f^Γ. The matrix K, however, will be singular when C^Γ has less than full rank. This condition occurs when the cut-joint constraints are not all independent. It can be shown that in such cases d will lie in the column space of K. Consequently, Eq. (44) will have an infinity of solutions for λ, and any of these can be used in Eq. (43) to establish f^Γ. Solving Eq. (44) for λ is an $\mathcal{O}(r^3)$ operation, where r is the total number of constrants of all cut joints. In most problems, however, the number of cut joints with constraints is small, and hence this operation will rarely have a major impact on program running speed.

For computational efficiency in implementations, it is worthwhile to write out the preceding equations for establishing f^Γ in terms of submatrices and subvectors. Equations (12) and (14) show that each row block α of C^Γ contains only one or two nonzero blocks, namely, $C^\Gamma_{\alpha j_{\alpha_I}}$ and $C^\Gamma_{\alpha j_{\alpha_O}}$. This enables Eqs. (43), (45), and (47) to be rewritten as

$$f_i^\Gamma = \sum_{\gamma \in S_{\Gamma i}} C_{\gamma i}^{\Gamma T} \lambda_\gamma \tag{48}$$

$$d_\alpha = -\sum_{k=j_{\alpha_I}, j_{\alpha_O}} \left\{ \dot{C}^\Gamma_{\alpha k} v_k + C^\Gamma_{\alpha k} \dot{w}_k \right\} \tag{49}$$

$$H_{\alpha i} = C^\Gamma_{\alpha j_{\alpha_I}} A_{j_{\alpha_I} i} + C^\Gamma_{\alpha j_{\alpha_O}} A_{j_{\alpha_O} i} \tag{50}$$

where $i = 1$ to n and $\alpha = 1$ to m, where m is the number of constrained cut joints. The $S_{\Gamma i}$ is the integer set containing the numbers of the constrained cut joints that are attached to Z_i. Since $A_{ji} = 0$ for $i \notin S_{Ij}$, it is evident that $H_{\alpha i} = 0$ for $i \notin S_{Ij_{\alpha_I}} \cup S_{Ij_{\alpha_O}}$. When calculating the A_{ji} for use in Eq. (50) to determine the $H_{\alpha i}$, it is not necessary to establish the full matrix A. Rather, one needs to compute only the row blocks of A that pertain to bodies to which cut joints with constraints are attached. The element blocks of each such row block can be computed sequentially moving left from the diagonal and employing the matrices $A_{i,i-1}$, which are established by Eq. (31). The algorithm is

$$A_{ji} = A_{j,i+1(j)} A_{i+1(j),i} \tag{51}$$

for $i = j - 1$ to 1. After the $H_{\alpha i}$ have been computed, the blocks $K_{\alpha\gamma}$ of K can be established via

$$K_{\alpha\gamma} = \sum_{i \in S_{\alpha\gamma}} H_{\alpha i} N_{ii} H_{\gamma i}^T \tag{52}$$

where $S_{\alpha\gamma} = (S_{Ij_{\alpha_I}} \cup S_{Ij_{\alpha_O}}) \cap (S_{Ij_{\gamma_I}} \cup S_{Ij_{\gamma_O}})$.

Summary of the Algorithm

An algorithm for using the preceding equations to compute the $\dot{v}_i'$ vectors is summarized in this section. The algorithm assumes that the system state at the computation time is available from the preceding integration step. The operations are as follows:

1) Do a forward pass over Z. For each Z_i, first compute $R_{i,i-1}$ and T_{ii} and then compute v_i by Eq. (6) with initial condition $v_O = \mathbf{0}$.

2) Do a backward pass over Z. For each Z_i, first compute M_i and f_i and then compute M_i^*, M_i', f_i^*, b_i, N_{ii}', a_i', $A_{i,i-1}'$, $A_{i,i-1}$, a_i, and g_i using Eqs. (34), (36), (20), (8), (29), (28), (27), (31), (32), and (35), respectively.

3) Do a forward pass over Z. Compute $\dot{w}_i$ and N_{ii} by Eqs. (42) and (33). Also, employ Eq. (51) to compute the A_{ji} matrices needed later for calculating the $H_{\alpha i}$.

4) Do a pass over the constrained cut joints Γ. For each Γ_α, compute $C^\Gamma_{\alpha j_{\alpha_I}}$, $C^\Gamma_{\alpha j_{\alpha_O}}$, $\dot{C}^\Gamma_{\alpha j_{\alpha_I}} v_{j_{\alpha_I}}$, and $\dot{C}^\Gamma_{\alpha j_{\alpha_O}} v_{j_{\alpha_O}}$. Also, compute d_α by Eq. (49). Also, do an inner pass over Z to compute the $H_{\alpha i}$ by Eq. (50).

5) Do a pair of nested forward passes over Γ to compute K by Eq. (52).

6) Solve Eq. (44) for λ.

7) Do a backward pass over Z. For each i, first compute f_i^Γ by Eq. (48) and then compute $f_i^{\Gamma^*}$ by Eq. (37).

8) Do a forward pass over Z. Compute $\dot{v}_i'$ and $\dot{v}_i$ by Eqs. (25) and either (26) or (7) with initial condition $\dot{v}_O = \mathbf{0}$.

Steps 3–7 are omitted if Σ' has no cut joints with constraints.

Example

A sample problem is presented next to illustrate the technique. Consider an unattached system of two bodies interconnected by a pair of constrained joints. Cut one of the joints to break the closed loop. Using the approach developed in the preceding sections, the equations for calculating $\dot{v}_1'$ and $\dot{v}_2'$ can be determined to be as follows:

$$v_1 = T_{11} v_1' \tag{53}$$

$$v_2 = R_{21} v_1 + T_{22} v_2' \tag{54}$$

$$M_2^* = M_2 \tag{55}$$

$$M_2' = T_{22}^T M_2^* T_{22} \tag{56}$$

$$f_2^* = f_2 \tag{57}$$

$$b_2 = \dot{R}_{21} v_1 + \dot{T}_{22} v_2' \tag{58}$$

$$N_{22}' = M_2'^{-1} T_{22}^T \tag{59}$$

$$a_2' = N_{22}' \left\{ f_2^* - M_2^* b_2 \right\} \tag{60}$$

$$A_{21}' = N_{22}' M_2^* R_{21} \tag{61}$$

$$A_{21} = R_{21} + T_{22} A_{21}' \tag{62}$$

$$a_2 = b_2 + T_{22} a_2' \tag{63}$$

$$g_2^* = M_2^* a_2 \tag{64}$$

$$M_1^* = M_1 + R_{21}^T M_2^* A_{21} \tag{65}$$

$$M_1' = T_{11}^T M_1^* T_{11} \tag{66}$$

$$f_1^* = f_1 + R_{21}^T f_2^* \tag{67}$$

$$b_1 = \dot{T}_{11} v_1' \tag{68}$$

$$N_{11}' = M_1'^{-1} T_{11}^T \tag{69}$$

$$a_1' = N_{11}' \left\{ f_1^* - M_1^* b_1 \right\} \tag{70}$$

$$a_1 = b_1 + T_{11} a_1' \tag{71}$$

$$\dot{w}_1 = a_1 \tag{72}$$

$$N_{11} = T_{11} N_{11}' \tag{73}$$

$$\dot{w}_2 = A_{21} \dot{w}_1 + a_2 \tag{74}$$

$$N_{22} = T_{22} N_{22}' \tag{75}$$

$$d = d_1 = -\sum_{j=2}^{2} \left\{ \dot{C}_{1j}^\Gamma v_j + C_{1j}^\Gamma \dot{w}_j \right\} \tag{76}$$

$$H_{11} = C_{11}^\Gamma + C_{12}^\Gamma A_{21} \tag{77}$$

$$H_{12} = C_{12}^\Gamma \tag{78}$$

$$K = K_{11} = \sum_{i=1}^{2} H_{1i} N_{ii} H_{1i}^T \tag{79}$$

$$K\lambda = d \qquad \text{(solve for } \lambda\text{)} \tag{80}$$

$$f_2^{\Gamma^*} = f_2^\Gamma = C_{12}^{\Gamma T} \lambda \tag{81}$$

$$f_1^\Gamma = C_1^{\Gamma T} \lambda \tag{82}$$

$$f_1^{\Gamma^*} = f_1^\Gamma + A_{21}^T f_2^{\Gamma^*} \tag{83}$$

$$\dot{v}_1' = a_1' + N_{11}' f_1^{\Gamma^*} \tag{84}$$

$$\dot{v}_1 = T_{11} \dot{v}_1' + b_1 \tag{85}$$

$$\dot{v}_2' = A_{21} \dot{v}_1 + a_2' + N_{22}' f_2^{\Gamma^*} \tag{86}$$

If the cut joint is a pin joint, then K is 5×5, and calculating λ, via Eq. (80), requires solving a fifth-order set of algebraic

equations. If the two bodies are rigid, then M_1 and M_2 are both 6×6, and the algorithm thus requires inverting a pair of 6×6 matrices; the most common algorithms employed in multibody dynamics would, instead, require the more time-consuming operations of calculating and inverting a single 12×12 generalized mass matrix. However it is not claimed, the approach presented in this paper is the most efficient one for systems that contain only two bodies. The superiority of the technique does not manifest itself until the number of bodies is large.

Conclusions

A new technique has been presented for computing the acceleration terms that are to be integrated numerically in multibody-dynamics computer programs. The technique permits the bodies to be rigid or nonrigid. The joints can have between 0 and 6 degrees of freedom. The system can be free in space or restrained relative to a fixed base. It can have open-chain, tree, or closed-loop topology. Closed loops are handled by the concept of cut joints. Constraint forces at cut joints are calculated by the Lagrange-multiplier aproach. Constraint forces at uncut joints do not appear in the formulation.

The technique is a member of the so-called order n class of multibody-dynamics formulations. It features sequential passes over the system of bodies instead of formulating and inverting a single large kinetics equation for the complete system. Programs employing the technique should run significantly faster than conventional multibody-dynamics programs when the number of bodies in the system being simulated is large.

References

[1]Sherman, M., "SD/EXACT and SD/FAST Symbolic Multibody Codes," *Proceedings of the SDIO/NASA Symposium on Multibody Simulations,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 690–728.

[2]Rodriguez, G., *Kalman Filtering, Smoothing and Recursive Robot-Arm Forward and Inverse Dynamics,* Jet Propulsion Lab., Pasadena, CA, JPL Pub. 86-48, Dec. 1986.

[3]Rodriguez, G., "Recursive Multirigid Body Topological Tree Dynamics via Kalman Filtering and Bryson-Frazier Smoothing," *Proceedings of the Sixth VPI & SU/AIAA Symposium on Control of Large Structures,* edited by L. Meirovitch, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, June 1987, pp. 45–60.

[4]Rodriguez, G., "Recursive Forward Dynamics for Two Robot Arms in a Closed Chain Based on Kalman Filtering and Bryson-Frazier Smoothing," *Robotics,* edited by M. Jamshidi, North-Holland, 1986.

[5]Rodriguez, G., "Spatially Recursive Filtering and Smoothing Methods for Multibody Dynamics," *Proceedings of the SDIO/NASA Workshop on Multibody Simulations,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 1094–1121.

[6]Rosenthal, D., "Order N Formulations for Equations of Motion of Multibody Systems," *Proceedings of the SDIO/NASA Workshop on Multibody Simulations,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 1122–1150.

[7]Haug, E. J., "Parallel Processing Real-Time Dynamic Simulation," *Proceedings of the SDIO/NASA Workshop on Multibody Simulations,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 1019–1070.

[8]Schwertassek, R., "Recursive Generation of Multibody System Equations," *Proceedings of the SDIO/NASA Workshop on Multibody Simulations,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 1151–1174.

[9]Keat, J. E., "Dynamical Equations of Multibody Systems with Application to Space Structure Deployment," Ph.D. Dissertation, Massachusetts Inst. of Technology, Cambridge, MA, May 1983.

[10]Keat, J. and Turner, J., "Equations of Motion of Multibody Systems for Application to Large Space Structure Deployment," AIAA Paper 84-1014, May 1984.

[11]Chun, H., Keat, J., Messac, A., and Turner, J., "Large Space Structure Deployment Dynamics Interim Technical Report—Section 2, The Charles Stark Draper Lab., Cambridge, MA, CSDL-R-1722, Sept. 1984.

[12]Keat, J. and Turner, J., "Equations of Motion of Variable Mass Bodies for Space Structure Deployment Simulation," *Proceedings of the Sixth VPI & SU/AIAA Symposium on Dynamics and Control of Large Space Structures,* edited by L. Meirovitch, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, 1987, pp. 549–562.

[13]Keat, J. "A Formulation for Simulating Deployment Dynamics of Large Space Structures," *Proceedings of the SDIO/NASA Workshop on Multibody Simulation,* Jet Propulsion Lab., Pasadena, CA, Sept. 1987, pp. 418–442.

[14]Jerkovsky, W., "The Structure of Multibody Dynamics Equations," *Journal of Guidance and Control,* Vol. 1, May–June 1978, pp. 173–182.