SYSTEMATIC GENERATION OF MULTIBODY EQUATIONS OF MOTION SUITABLE FOR RECURSIVE AND PARALLEL MANIPULATION

Parviz E. Nikravesh, Gwanghun Gim, Ara Arabyan, Udo Rein

Computer-Aided Engineering Laboratory
Aerospace and Mechanical Engineering Department
University of Arizona
Tucson, AZ 85721

ABSTRACT

This paper summarizes the formulation of a method known as the joint coordinate method for automatic generation of the equations of motion for multibody systems. For systems containing open or closed kinematic loops, the equations of motion can be reduced systematically to a minimum number of second order differential equations. The application of recursive and nonrecursive algorithms to this formulation, computational considerations and the feasibility of implementing this formulation on multiprocessor computers are discussed.

1. INTRODUCTION

In the past decade, the joint (or natural) coordinate method has been implemented in formulating the equations of motion. The methodology for open loop systems is well developed in a variety of forms [1-5]. For these systems, the method yields a minimal set of equations of motion since the joint coordinates are independent. The joint coordinates are no longer independent when closed kinematic loops exist in a system. For multibody systems containing simple closed loops, constraint equations between joint coordinates may be derived easily. However, for most spatial closed loops, the derivation of these constraints can be rather complicated. A simple method for automatic generation of the closed loop constraints, and a technique to generate a minimal set of differential equations of motion has been reported in reference [6].

This paper briefly describes the method of joint coordinates for the systematic generation of the equations of motion for open and closed loop systems. These equations are shown to be suitable for recursive and nonrecursive algorithms, serial or parallel processing, and symbolic manipulation. While the discussion principally focuses on multi-rigid-body systems, the assumption of rigidity may be relaxed by introducing the finite element technique in modeling the deformation of bodies. This formulation has been incorporated in a set of large-scale computer programs which have been used extensively to simulate the dynamic behavior of a variety of multibody systems.

2. EQUATIONS OF MOTION

The equations of motion for a multibody mechanical system can be expressed in different forms depending upon the choice of the coordinates and velocities that describe the configuration and motion of the system. In the following subsections, the equations of motion are first expressed in terms of absolute coordinates and velocities of the bodies in the system. Then these equations are reduced to a minimal set of equations for open-loop systems. Furthermore, the equations are generalized for systems containing closed kinematic loops.

2.1 Absolute Coordinate Formulation

In order to specify the position of a rigid body in a global non-moving xyz coordinate system, it is sufficient to specify the spatial location of the origin (center of mass) and the angular orientation of a body-fixed $\xi\eta\zeta$ coordinate system as shown in Fig. 1. For the ith body in a multibody system, vector q_i denotes a vector of coordinates which contains a vector of Cartesian translational coordinates r_i and a set of rotational coordinates. Matrix A_i represents the rotational transformation of the $\xi\eta\zeta_i$ axes relative to the xyz axes. A vector of velocities for body i is defined as v_i, which contains a 3-vector of translational velocities $\dot{r}_i$ and a 3-vector of angular velocities ω_i. The components of the angular velocity vector ω_i are defined in the xyz coordinate system rather than the body-fixed coordinate system. A vector of accelerations for this body is denoted by $\dot{v}_i$, which contains $\ddot{r}_i$ and $\dot{\omega}_i$. For a multibody system containing b bodies, the vectors of coordinates, velocities, and accelerations are q, v, and $\dot{v}$ which contain the elements of q_i, v_i, and $\dot{v}_i$, respectively, for i = 1, ..., b.

The kinematic joints between the bodies can be described by m-independent holonomic constraints as

$$\Phi(q) = 0 \tag{1}$$

The first and second time derivatives of the constraints yield the kinematic velocity and acceleration equations

$$\dot{\Phi} \equiv Dv = 0 \tag{2}$$

$$\ddot{\Phi} \equiv D\dot{v} + \dot{D}v = 0 \tag{3}$$

where D is the Jacobian matrix of the constraints. The equations of motion are written as [7]

$$M\dot{v} - D^T\lambda = g \tag{4}$$

where M is the inertia matrix containing the mass and the inertia tensor of all bodies, λ is a vector of m Lagrange multipliers, and $g = g(q, v)$ contains the gyroscopic terms and the forces and moments that act on the system. The inertia matrix is not a constant matrix since the rotational equations of motion are written in terms of the global components of the angular accelerations. The term $D^T\lambda$ in Eq. 4 represents the joint reaction forces and moments. Equations 1-4 represents a set of differential-algebraic equations of motion for a multibody system when absolute coordinates are used. These equations will have the same form whether the system is open- or closed-loop.

2.2 Joint Coordinates and Open Loop Systems

The constrained equations of motion expressed by Eqs. 1-4 can be converted to a smaller set of equtions in terms of a set of coordinates known as joint coordinates. For multibody systems with open kinematic loops, this conversion yields a set of differential equations equal to the number of degrees of freedom. We may consider one branch of an open-loop system as shown in Fig. 2. The bodies are numbered in any desired order. The relative configurations of two adjacent bodies are defined by one or more so-called joint (or natural) coordinates equal in number to the number of relative degrees of freedom between these bodies. For example, θ_{ij} contains two angles if there is a universal joint between bodies i and j, or it contains only one translational variable if there is a prismatic joint between the two bodies. The vector of joint coordinates for the system is denoted by θ containing all of the joint coordinates and the absolute coordinates of a base (reference) body if the base body is not the ground (floating base). Therefore, vector θ has a dimension equal to the number of degrees of freedom of the system. The vector of joint velocities is defined as $\dot{\theta}$, which is the time derivative of θ. It can be shown that there is a linear transformation between $\dot{\theta}$ and v as [1-4]
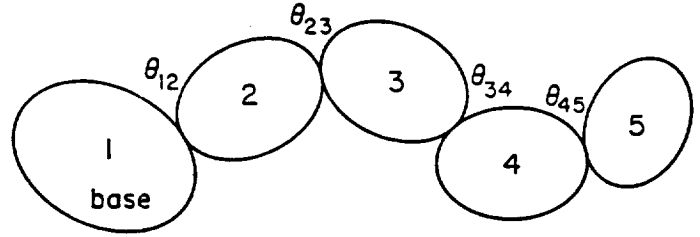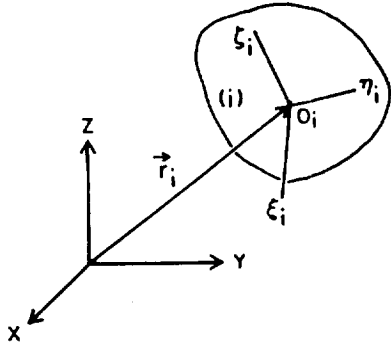
$$v = B\dot{\theta} \tag{5}$$

Figure 1 Body-fixed and global coordinate systems.   Figure 2 An open-loop system.

Matrix $\mathbf{B}$ is orthogonal to the Jacobian matrix $\mathbf{D}$. This can be shown by substituting Eq. 5 in Eq. 2 to find $\mathbf{DB}\dot{\theta} = 0$. Since $\dot{\theta}$ is a vector of independent velocities, then

$$\mathbf{DB} = 0 \tag{6}$$

The time derivative of Eq. 5 gives the transformation formula for the accelerations,

$$\dot{\mathbf{v}} = \mathbf{B}\ddot{\theta} + \dot{\mathbf{B}}\dot{\theta} \tag{7}$$

Substituting Eq. 7 in Eq. 4, premultiplying by $\mathbf{B}^\mathsf{T}$, and using Eq. 6 yields

$$\bar{\mathbf{M}}\ddot{\theta} = \mathbf{f} \tag{8}$$

where

$$\bar{\mathbf{M}} = \mathbf{B}^\mathsf{T}\mathbf{MB} \tag{9}$$

$$\mathbf{f} = \mathbf{B}^\mathsf{T}(\mathbf{g} - \mathbf{M}\dot{\mathbf{B}}\dot{\theta}) \tag{10}$$

Equation 8 represents the generalized equations of motion for an open-loop multibody system when the number of the selected coordinates is equal to the number of degrees of freedom.

2.3 Joint Coordinates and Closed-Loop Systems [6]

The equations of motion for open-loop systems, represented by Eq. 8, can be modified for systems containing closed kinematic loops. Assume that there is one or more closed kinematic loops in a multibody system, such as the closed-loop shown in Fig. 3(a). In order to derive the equations of motion for such a system, the closed-loop is cut at one of the kinematic joints in order to obtain an open-loop system as shown in Fig. 3(b). For this cut system, which will be called the reduced system, the joint coordinates are defined as for any open-loop system. It is clear that if we now close this system at the cut joint(s), the joint coordinates will no longer be independent, i.e., for each closed-loop there exist one or more algebraic constraints between the joint coordinates of that loop.

The constraint equations for the closed kinematic loops may be expressed implicitly as

$$\mathbf{\Upsilon}(\theta) = 0 \tag{11}$$

The time derivative of the constraints yields

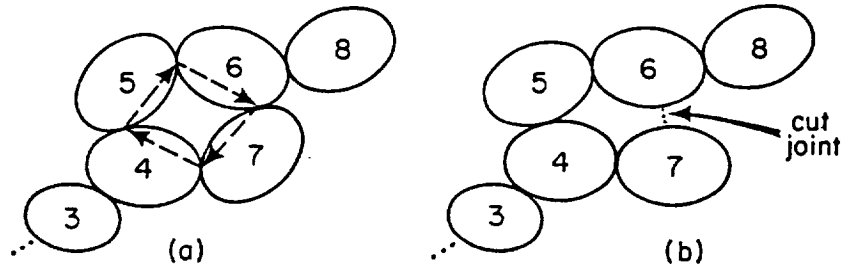$$\dot{\mathbf{\Upsilon}} \equiv \mathbf{C}\dot{\theta} = 0 \tag{12}$$

Figure 3 A system containing a closed-loop. (a) The closed-loop.
(b) Its reduced open-loop representation.

where C is the Jacobian matrix of these constraints. Similarly the acceleration constraints for the closed-loops are written as

$$\ddot{\mathbf{Y}} \equiv \mathbf{C}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{C}}\dot{\boldsymbol{\theta}} = 0 \tag{13}$$

Now the equations of motion of Eq. 8 are modified for closed loop systems as

$$\bar{\mathbf{M}}\ddot{\boldsymbol{\theta}} - \mathbf{C}^T \mathbf{v} = \mathbf{f} \tag{14}$$

where $\mathbf{v}$ is a vector of Lagrange multipliers associated with the constraints of Eq. 11. Equations 11-14 represent the equations of motion for a multibody system when the number of selected joint coordinates is greater than the number of degrees of freedom of the system.

It is shown in [6] that the Jacobian matrix C in Eqs. 12-14 can be obtained systematically. If the Jacobian of the constraints for the cut-joint(s) is denoted by $\mathbf{D}^*$, then the product of this matrix and matrix B yields the C matrix as

$$\mathbf{D}^* \mathbf{B} \rightarrow \mathbf{C} \tag{15}$$

The product $\mathbf{D}^* \mathbf{B}$, in most cases, will have redundant rows. Matrix C is found after the redundant rows are eliminated. Since the elements of both matrices $\mathbf{D}^*$ and B are available explicitly in symbolic form, the elements of C can also be determined symbolically. Note that C is a small matrix in comparisson with $\mathbf{D}^*$ and B. Furthermore, since the elements of C can be determined symbolically, the term $\dot{\mathbf{C}}\dot{\boldsymbol{\theta}}$ in Eq. 13 can also be found symbolically.

2.4 Minimum Number of Equations of Motion for Closed-Loop Systems

For a multibody system containing closed kinematic loops, the Lagrange multipliers of Eq. 14 can be eliminated in order to obtain a minimal set of equations of motion in terms of a set of independent joint accelerations. For this purpose, a set of independent joint coordinates are selected as a subset of vector $\boldsymbol{\theta}$. Then a velocity transformation matrix E can be found such that [6]

$$\dot{\boldsymbol{\theta}} = \mathbf{E}\dot{\boldsymbol{\theta}}_{(i)} \tag{16}$$

where $\dot{\boldsymbol{\theta}}_{(i)}$ is the vector of independent joint velocity with a dimension equal to the number of degrees of freedom of the system. Note that the joint velocities outside the closed-loops and the independent joint velocities within the closed-loops are not affected by this conversion. The matrix E is orthogonal to the C matrix; i.e.,

$$\mathbf{C}\mathbf{E} = 0 \tag{17}$$

The time derivative of Eq. 17 gives

46

$$\ddot{\theta} = E\ddot{\theta}_{(i)} + \dot{E}\dot{\theta}_{(i)} \tag{18}$$

Substituting of Eq. 18 in Eq. 14, premultiplying by E^T, and using Eq. 16 yields

$$\bar{M}'\ddot{\theta}_{(i)} = f' \tag{19}$$

where

$$\bar{M}' = E^T\bar{M}E \tag{20}$$

$$f' = E^T(f - \bar{M}E\dot{\theta}_{(i)}) \tag{21}$$

Equation 19 represents the minimum number of equations of motion describing the dynamics of a multibody system containing closed kinematic loops.

Matrix E can be found in either explicit form or in numerical form for most closed kinematic loops. For this purpose, we can partition vector $\dot{\theta}$ into dependent and independent sets, $\dot{\theta}_{(d)}$ and $\dot{\theta}_{(i)}$, and correspondingly we can partition matrix C into two submatrices $C_{(d)}$ and $C_{(i)}$. Therefore, Eq. 16 becomes

$$C_{(d)}\dot{\theta}_{(d)} + C_{(i)}\dot{\theta}_{(i)} = 0$$

This equation can be written as,

$$\dot{\theta}_{(d)} = - C_{(d)}^{-1}C_{(i)}\dot{\theta}_{(i)}$$

or,

$$\dot{\theta} = \begin{bmatrix} I \\ -C_{(d)}^{-1}C_{(i)} \end{bmatrix} \dot{\theta}_{(i)} \tag{22}$$

where a proper selection of the independent joint velocities guarantees that $C_{(d)}$ is a nonsingular matrix. Comparing Eqs. 16 and 22 yields an expression for E as

$$E = \begin{bmatrix} I \\ -C_{(d)}^{-1}C_{(i)} \end{bmatrix} \tag{23}$$

Since the elements of matrix C are available explicitly, it may be possible to find the elements of E explicitly. This is due to the fact that the operation of Eq. 23 is performed separately on each closed-loop, and in addition, the C matrix for a closed-loop is relatively small in practical applications.

Equation 21 states that for evaluating the equations of motion, in addition to matrix E, matrix $\dot{E}$ is also needed. An apparent approach is to evaluate the time derivative of Eq. 23. However, since the product $\dot{E}\dot{\theta}_{(i)}$ is needed in Eq. 21, we can evaluate this product directly. For this purpose, Eq. 13 is written in a partitioned form as

$$C_{(d)}\ddot{\theta}_{(d)} + C_{(i)}\ddot{\theta}_{(i)} = -\dot{C}\dot{\theta}$$

This equation is then rearranged as

$$\ddot{\theta}_{(d)} = - C_{(d)}^{-1}C_{(i)}\ddot{\theta}_{(i)} - C_{(d)}^{-1}\gamma\dot{C}\dot{\theta}$$

or,

$$\ddot{\theta} = \begin{bmatrix} I \\ -C_{(d)}^{-1} C_{(i)} \end{bmatrix} \ddot{\theta}_{(i)} + \begin{bmatrix} 0 \\ -C_{(d)}^{-1} \dot{C} \dot{\theta} \end{bmatrix} \tag{24}$$

Comparison of Eqs. 18 and 24 yields

$$E \ddot{\theta}_{(i)} = \begin{bmatrix} 0 \\ -C_{(d)}^{-1} \dot{C} \dot{\theta} \end{bmatrix} \tag{25}$$

In the process of solving Eq. 19, the independent joint accelerations and velocities are integrated to obtain the independent joint velocities and coordinates. Then Eq. 16 (or Eq. 12) is used to find the dependent joint velocities. However, prior to that the dependent joint coordinates need to be computed. In order to find the dependent joint coordinates, the constraints of Eq. 11 must be solved for each closed-loop. These constraints are nonlinear in θ (or q) and they are not available explicitly. However, an iterative formula for finding the dependent joint coordinates can be derived. By applying the Newton-Raphson method to Eq. 11, the iterative formula is found as

$$C \, \Delta\theta = - \, \phi^* \tag{26}$$

where $\Delta\theta$ denotes the corrections in vector θ and ϕ^* denotes the violation in the constraints. Note that the violation in the constraints of Eq. 11 is the same as the violation in the constraints written in terms of the absolute coordinates of the bodies common to the cut joint(s). For the sake of simplicity, the iteration formula of Eq. 26 is shown in its abstract form. We assume it is understood that in this equation the known (independent) elements of θ and their corresponding columns of C have been manipulated properly. Furthermore, if there is more than one closed kinematic loop in the system, this iterative process can be applied separately to the constraint equations of each loop. The important point to draw from Eq. 26 is that explicit expressions for the joint coordinate constraints, represented by Eq. 11, are not needed.

3. COMPUTATIONAL CONSIDERATIONS

For most multibody systems, the inertia matrix M and the vector g containing the external forces/moments and the gyroscopic terms can be constructed systematically [7]. A systematic construction of the velocity transformation matrix B, for open or reduced open loop systems, has been shown in [4]. This matrix is constructed from the topology of the system by assembling smaller block matrices representing different type of kinematic joints. Since matrix B can be constructed in explicit form in terms of the absolute coordinates q, then matrix B can also be determined and expressed in explicit form as a function of q and v. For multibody systems containing closed kinematic loops, matrix C for Eqs. 11-14 and matrix E for Eq. 18 must be constructed. The process outlined in the preceding sections will be demonstrated by a simple example.

Example: Consider the multibody system shown in Fig. 4(a), containing four moving bodies. Body 1 is connected to the ground by a prismatic joint T_1, and there are four revolute joints, R_2 through R_5, with parallel axes connecting the bodies in a closed-loop. If the closed-loop is cut at R_5, the reduced system of Fig. 4(b) is obtained. For the reduced system, four joint coordinates, θ_1, θ_2, θ_3, and θ_4 are defined, where θ_1 represents relative translation between body 1 and the ground, and the other three joint coordinates represent relative rotations between the adjacent bodies. Four unit vectors, u_1 through u_4, are defined along the four joint axes. The vector of absolute velocities v is a 24-vector, where the vector of relative velocities $\dot{\theta}$ is a 4-vector. The velocity transformation matrix for this reduced system is:
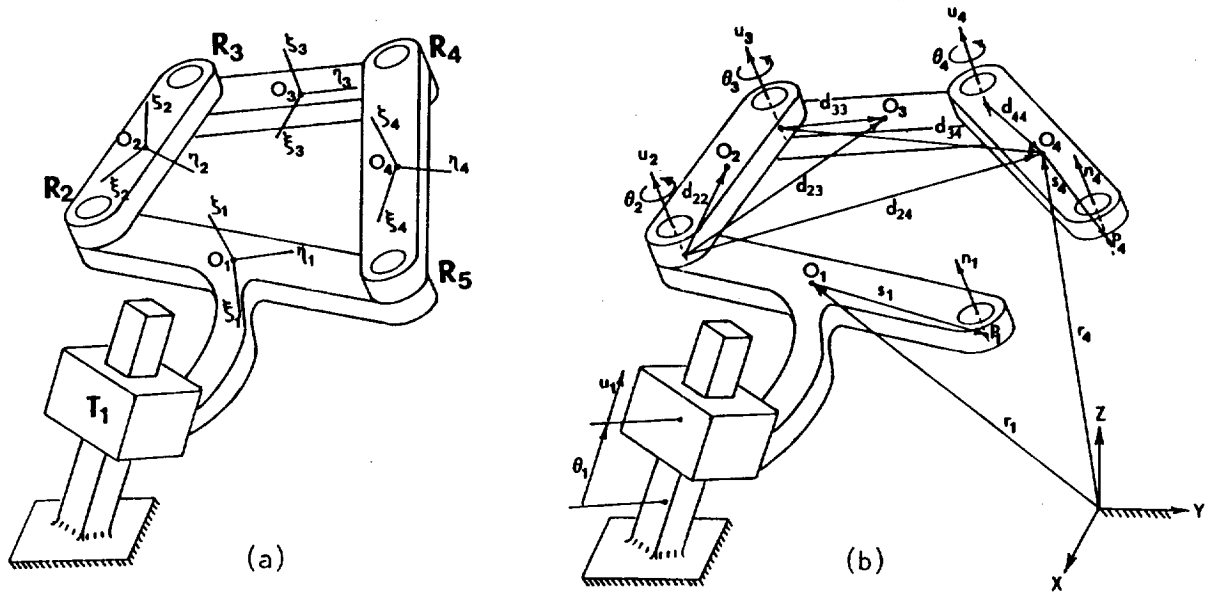
Figure 4 (a) A multibody system with four moving bodies containing a closed-loop.
(b) Its reduced open-loop representation.

$$
B = \begin{bmatrix}
u_1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
u_1 & -\tilde{d}_{22}u_2 & 0 & 0 \\
0 & u_2 & 0 & 0 \\
u_1 & -\tilde{d}_{23}u_2 & -\tilde{d}_{33}u_3 & 0 \\
0 & u_2 & u_3 & 0 \\
u_1 & -\tilde{d}_{24}u_2 & -\tilde{d}_{34}u_3 & -\tilde{d}_{44}u_4 \\
0 & u_2 & u_3 & u_4
\end{bmatrix}
$$

where d_{ij} vectors, for $i, j = 2, 3, 4$, are shown in Fig. 4(b). In this matrix, $\tilde{a}$ represents a 3x3 skew-symmetric matrix made of the components of a 3-vector a, and $\tilde{a}b$ represents the cross product of two vectors a and b. The structure of B shows that the matrix is constructed from 6x1 block matrices $\begin{bmatrix} u_1 \\ 0 \end{bmatrix}$, representing a prismatic joint along the unit vector u_1, and 6x1 block matrices $\begin{bmatrix} -\tilde{d}_{ij}u_i \\ u_i \end{bmatrix}$, representing revolute joints along unit vectors u_i, $i = 2, 3,$ and 4 [4].

The constraint equations for the cut revolute joint R_s, i.e., Φ^*, between bodies 1 and 4 in terms of the absolute coordinates of these bodies can be expressed as [7]:

$$\Phi^3 \equiv r_1 + s_1 - r_4 - s_4 = 0$$

$$\Phi^2 \equiv \tilde{n}_1 n_4 = 0$$

where Φ^3 represents three algebraic equations stating that two points P_1 and P_4 on the joint axis must coincide, and Φ^2 states that two unit vectors n_1 and n_4 along the axis of R_s must remain parallel. It should be noted that the cross product of two vectors yields three abgebraic constraints, and only two out of three are independent. The time derivative of these constraints yields [7]:

$$\dot{\phi}^* \equiv \begin{cases} \dot{r}_1 + \tilde{\omega}_1 s_1 - \dot{r}_4 - \tilde{\omega}_4 s_4 = 0 \\ \tilde{n}_4 \tilde{n}_1 \omega_1 - \tilde{n}_1 \tilde{n}_4 \omega_4 = 0 \end{cases}$$

or,

$$\begin{bmatrix} 1 & -\tilde{s}_1 & 0 & 0 & 0 & 0 & -1 & \tilde{s}_4 \\ 0 & \tilde{n}_4 \tilde{n}_1 & 0 & 0 & 0 & 0 & 0 & -\tilde{n}_1 \tilde{n}_4 \end{bmatrix} \begin{bmatrix} \dot{r}_1 \\ \omega_1 \\ \dot{r}_2 \\ \omega_2 \\ \dot{r}_3 \\ \omega_3 \\ \dot{r}_4 \\ \omega_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Therefore, the coefficients of the velocity components provide the Jacobian matrix for this cut revolute joint as:

$$D^* = \begin{bmatrix} 1 & -\tilde{s}_1 & 0 & 0 & 0 & 0 & -1 & \tilde{s}_4 \\ 0 & \tilde{n}_4 \tilde{n}_1 & 0 & 0 & 0 & 0 & 0 & -\tilde{n}_1 \tilde{n}_4 \end{bmatrix}_{(5 \times 24)}$$

Note that this is a 5x24 matrix. The product $D^* B$ is found to be

$$D^* B = \begin{bmatrix} 0 & (\tilde{d}_{24} + \tilde{s}_4) u_2 & (\tilde{d}_{34} + \tilde{s}_4) u_3 & (\tilde{d}_{44} + \tilde{s}_4) u_4 \\ 0 & -\tilde{n}_1 \tilde{n}_4 u_2 & -\tilde{n}_1 \tilde{n}_4 u_3 & -\tilde{n}_1 \tilde{n}_4 u_4 \end{bmatrix}_{(5 \times 4)}$$

which is a 5x4 matrix since the columns of the matrix correspond to the four joint coordinates. Note that the first column of the matrix is all zeros, and this column corresponds to θ_1 which is not in the closed-loop. Based on the initial assumption, the four revolute joint axes are parallel, therefore, the cross product $\tilde{n}_4 u_2 = \tilde{n}_4 u_3 = \tilde{n}_4 u_4 = 0$. This means that the last two rows of the 5x4 matrix are zeros and they can be eliminated from the matrix. This leaves a 3x4 matrix as:

$$D^* B = [0 \quad (\tilde{d}_{24} + \tilde{s}_4) u_2 \quad (\tilde{d}_{34} + \tilde{s}_4) u_3 \quad (\tilde{d}_{44} + \tilde{s}_4) u_4]_{(3 \times 4)}$$

If for a given configuration numerical values are substituted for the components of the vectors in this matrix, it will be found that the three rows are not independent -- one row is redundant and can be eliminated. For example, for a particular configuration, the numerical values of the elements of this matrix may be:

$$D^* B = \begin{bmatrix} 0 & 0 & -1.4142 & -1.4142 \\ 0 & 0 & -1.4142 & -1.4142 \\ 0 & -1.4142 & -1.4142 & 0 \end{bmatrix}_{(3 \times 4)} \tag{a}$$

This result should have been expected, since the closed-loop is a four-bar mechanism with one relative degree of freedom between its four bodies. Since there are three joint coordinates associated with this four-bar mechanism, there must be only two independent constraints between them. Therefore, matrix $D^* B$ of Eq. (a) becomes

$$C = \begin{bmatrix} 0 & 0 & -1.4142 & -1.4142 \\ 0 & -1.4142 & -1.4142 & 0 \end{bmatrix}_{(2\times4)} \qquad (b)$$

Since matrix C is available in explicit form, its time derivative and hence the product $\dot{C}\dot{\theta}$ can be found for Eq. 13.

After elimination of the redundant row matrix C can be written in abstract form as

$$C = \begin{bmatrix} 0 & c_1 & c_2 & c_3 \\ 0 & c_4 & c_5 & c_6 \end{bmatrix}$$

where c_1, ..., c_6 are known explicitly from Eq. (a). If we choose θ_2 as the independent joint coordinate for the closed-loop, and noting that θ_1 is independent from the loop, we can have

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = E \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

where

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & e_1 \\ 0 & e_2 \end{bmatrix}, \quad e_1 = \frac{c_3 c_4 - c_1 c_6}{c_2 c_6 - c_3 c_5}, \text{ and } e_2 = \frac{c_1 c_5 - c_2 c_4}{c_2 c_6 - c_3 c_5}$$

If the numerical values of the elements of C from Eq. (b) are used, matrix E will be found to be

$$E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \\ 0 & 1 \end{bmatrix}$$

4. RECURSIVE VERSUS NONRECURSIVE ALGORITHMS

The equations of motion for open or closed loop systems (Eq. 8 or Eqs. 11-14) can be generated and solved for the accelerations either recursively or nonrecursively. Since the inertia matrix $\bar{M}$ is symmetric, a standard nonrecursive technique such as $L^T DL$ factorization can be employed to solve for the unknown accelerations. An alternative recursive approach for finding the unknown accelerations was first developed by Featherstone [8]. The idea is to remove one body from a multibody system and then to consider the remaining system as a new multibody system. Articulated body inertias are the properties which make the remaining system act as the original one. The articulated inertias are calculated by projecting the mass and inertia of the removed body onto the remaining system. Repeatedly removing one body from the multibody system leads to a system with only one body for which the accelerations can easily be calculated. The accelerations of the removed bodies will then be obtained by back substitution. Wehage interpreted this process mathematically by using a large number of equations of motion [9-11]. The unknowns of the equations are the absolute and joint accelerations, as well as joint reaction forces. The equations of motion are solved by applying matrix partitioning. This more theoretical approach allows for a very general formulation of the recursive

51

projection algorithm. Wehage shows that a recursive algorithm is equivalent to a "block LU factorization."

For an open loop system containing n joints, a nonrecursive matrix factorization algorithm requires a CPU time of approximately Order(n^2). For the same system, a recursive algorithm requires a CPU time of Order(n). However, the factor in front of O(n) or O(n^2) can make one algorithm more or less efficient than the other, depending upon n. Therefore, some examples are shown in this section to clarify this concept.

In the recursive algorithms, the velocity transformation matrix B can be represented as the product of two matrices [9],

$$B = G^{-1}H \tag{27}$$

The matrices G and H are found from velocity transformation equations between consecutive bodies as

$$v_j = G_j v_i + H_j \dot{\theta}_j \tag{28}$$

The equations of motion for an open loop system; i.e., Eq. 8, are then written as

$$(G^{-1}H)^T M G^{-1} H \ddot{\theta} = (G^{-1}H)^T (g - MG^{-1}\gamma) \tag{29}$$

where γ contains quadratic velocity terms which can be constructed from

$$\gamma_j = \dot{G}_j v_i + \dot{H}_j \dot{\theta}_j \tag{30}$$

A detailed description of the recursive algorithm used in this study can be found in [12].

Two simple examples are considered here for comparison of the CPU times. Figure 5 shows a highly parallel and a highly serial system. In both systems, the number of joints n is increased, starting from one, between simulations. For the parallel system with only revolute joints, the nonrecursive method is more efficient than the recursive method regardless of number of joints, as shown in Fig. 6(a). It can be observed that both algorithms yield CPU times that increase almost linearly as n is increased. For the serial system, however, there is a breakpoint beyond which the recursive algorithm becomes more efficient than the nonrecursive algorithm. As shown in Fig. 6(b, c, d), the breakpoint for the number of joints n is six, nine, and five when the serial system contains only revolute joints, prismatic joints, or spherical joints respectively.

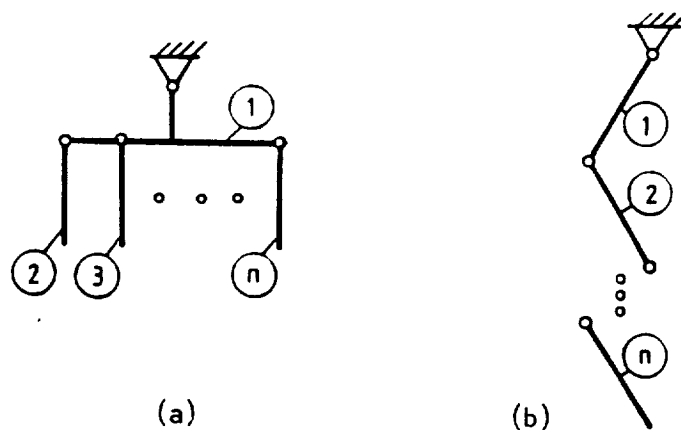

(a)   (b)

Figure 5 Two systems with (a) a highly parallel and (b) a highly serial structures.
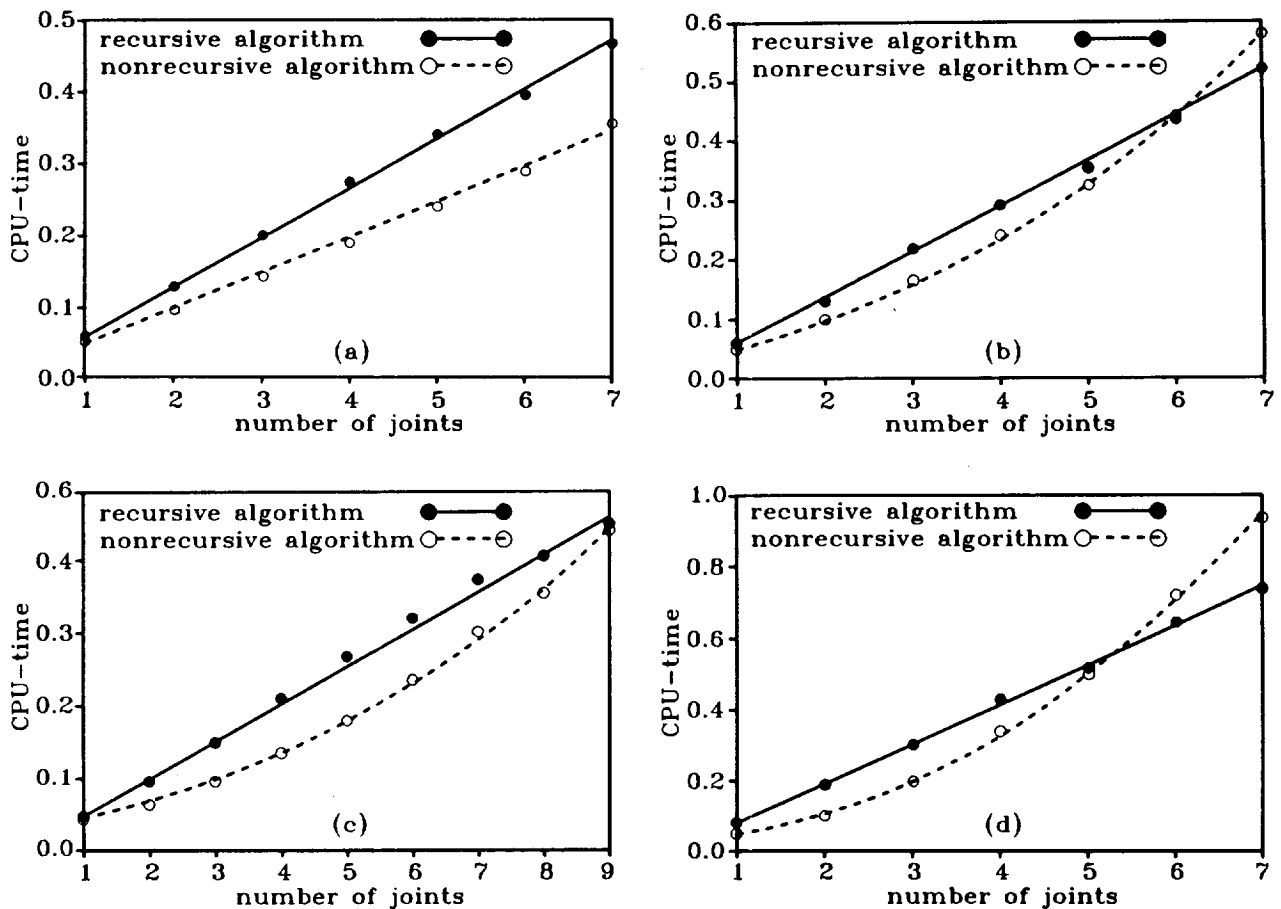
Figure 6 CPU time for one function evaluation versus number of joints for (a) a highly parallel system, and (b), (c), (d) highly serial systems containing only revolute, prismatic, and spherical joints respectively.

In reference [11], the recursive projection algorithm of open loop systems is modified for systems containing closed kinematic loops. This algorithm has been tested and the result is reported in [12]. It is shown that since a closed loop is cut at one of the joints to form a reduced open loop system, the breakpoint for the number of joints in a closed loop is approximately double of that of an open loop system. For example, if the closed loop contains only revolute joints, there should be approximately twelve or more bodies in the loop for the recursive algorithm to exhibit more efficiency than the nonrecursive method.

For mechanical systems with only rigid bodies, it is rather unlikely to have open or closed loops with enough number of bodies to make a recursive algorithm more efficient than a nonrecursive one. However, when one or more of the bodies in a system are considered as deformable, then the concept of recursive projection technique becomes highly attractive.

5. PARALLEL COMPUTATIONAL CONSIDERATIONS

When computation on a multiple-instruction multiple-data (MIMD) multiprocessing system is considered, obvious parallelisms arising from the topology (e.g. multiple branches) can be exploited. However, a true measure of the suitability of a computational scheme for parallel processing is the degree of intrinsic parallelism in the scheme for the worst case (i.e. single branch open-loop linkage).

53

The formulation described by Eq. 8 was applied to a 6 degree-of-freedom Stanford arm that consists of a base body plus 6 links all of which are connected to each other serially by revolute joints except link 3 which is connected to link 2 by a prismatic joint. The algorithm to compute θ at each time step was represented as a data-flow graph. It is known that the shortest possible time to traverse the graph from its beginning to its end is the length of the longest possible path in the graph, or the critical path.

The maximum speedup for any multiprocessing system is, therefore, the ratio of the serial computation time to the time corresponding to the critical path of the data-flow graph, since the latter is a property of the computational scheme alone. The length of the critical path was computed for a Stanford arm possessing 1 through 6 degrees of freedom. A plot of the maximum speedup with the degrees of freedom in the Stanford arm is shown in Fig. 7. The plot indicates the possibility of large speedups (11 to 76) if a suitable multiprocessing system and a proper scheduling algorithm are used. It is also observed that the length of the critical path of the data-flow graph resulting from the formulation increases linearly as the number of degrees of freedom increases in a serially connected multibody system. This suggests that the maximum speedup will approach a constant (ratio of the rate of increase of serial computation time to the rate of increase of critical path length) for a large number of degrees of freedom.
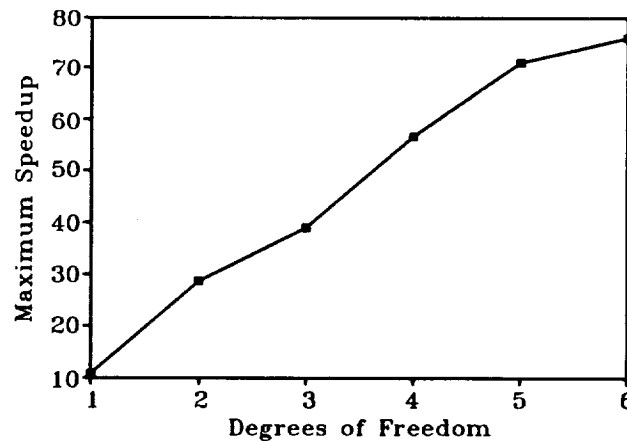

Figure 7 Maximum speedup versus number of degrees of freedom for the Stanford arm.

The velocity transformation of Eq. 5 offers other possibilities for parallel processing. It is evident from the approach described above that if the coordinate set, θ, used to describe the configuration of the system is such that the critical path of the data-flow graph does not increase or increases very slowly as new degrees of freedom are added to the system, then the maximum speedup will increase linearly or approach a very high limit at large numbers of degrees of freedom. Therefore, if matrix B can be chosen such that the resulting θ is this type of a coordinate set, then the formulation would be very suitable for parallel processing because of very large potential gains in speedup. Work is currently under way to determine such matrices B automatically on the basis of the topology imposed by each type of joint in a multibody system.

6. DEFORMABLE BODIES

In the dynamic analysis of multibody systems, the elastodynamic effects may play an important role on the behavior of the system. The most popular technique for describing the flexibility of the components of a system is the finite element method. In standard finite element formulation, the gross motion (large displacements, large deformations) is not taken into account. However, in order to analyze flexible multibody systems, such phenomena must be considered. Several researchers have suggested procedures that successfully introduce elastodynamic effects into multibody dynamics

equations [13-15]. The main problem with the inclusion of elastodynamic effects is that the flexible bodies may have a relatively complex geometry. This implies that a large number of nodes may be necessary and, therefore, a system of equations with a large number of degrees of freedom will result. In order to gain computational efficiency, a formulation based on the modal superposition method has been suggested to reduce the number of degrees of freedom of the model [14].

The method of joint coordinates for multi-rigidbody dynamics can be extended to a system of mixed rigid and flexible bodies. The equations of motion for the rigid bodies are written in terms of the absolute coordinates, similar to Eqs. 1-4, and the equations of motion for the flexible bodies are written in term of either nodal or modal coordinates. Additional kinematic constraints may be necessary to represent the connectivities between rigid-to-flexible and flexible-to-flexible bodies. Then, a velocity transformation process, similar to that of rigid bodies. (Eqs. 5-10) can be applied to remove the algebraic constraints and their corresponding Lagrange multipliers. The resultant equations of motion for an open loop system can be converted to a minimal set of differential equations. In the case of closed loop systems, the equations of motion may or may not contain algebraic constraints and Lagrange multipliers.

For systems containing flexible bodies with linear elastic material properties, the equations of motion with modal coordinates can be used. However, in some applications, it may be necessary to consider the equations of motion in terms of the nodal coordinates in order to obtain more accurate results.

7. CONCLUSION

The equations of motion for multibody systems containing closed kinematic loops can be written either as a set of differential-algebraic equations (Eq. 11-14) or as a set of ordinary differential equations (Eq. 19). The elements of the constrained equations of motion given by Eqs. 11-14 can be constructed efficiently. Although all of the joint coordinates are not independent of each other, and hence the number of integration variables is not a minimum, the numerical integration of these equations can be performed efficiently. For example, a dynamic simulation of the system shown in Fig. 4, including several force elements, was performed using two different formulations -- the absolute coordinate formulation of Eqs. 1-4 and the joint coordinate formulation of Eqs. 11-14. The numerical integration of the equations of motion in both cases was carried out using a predictor-corrector Adams-Bashforth algorithm on a desktop workstation. The CPU time for simulating ten seconds of dynamic response was 352 seconds for Eqs. 1-4 and 75 seconds for Eqs. 11-14. The results obtained from these and other simulations have shown that the formulation of Eqs. 11-14 yields about five times or more efficiency over the formulation of Eqs. 1-4. The degree of efficiency depends on the number of bodies, number of joints, and the connectivity between the bodies.

Equation 19 provides the minimum number of equations of motion for a multibody system containing closed kinematic loops. The number of equations and the number of integration variables are smaller when compared to those of Eqs. 11-14. Therefore, it may appear that the numerical solution of Eq. 19 to be more efficient than that of Eqs. 11-14. However, a careful examination of the elements of Eq. 19 would reveal that the overhead associated with evaluating these elements may be more than the overhead associated with the additional number of equations and integration variables of Eqs. 11-14. Numerical simulations of several problems using the two methods have shown that the computation time associated with these two formulations are about the same.

Systematic generation of the elements of the equations of motion with the joint coordinates makes the formulation ideal for symbolic generation of these elements. Computer programs have been developed that symbolically generate the equations of motion for rigid body systems. The equations are generated in an optimized fashion to improve the computational efficiency of the dynamic simulation. The programs dealing with rigid and flexible bodies evaluate the equations of motion numerically. The equations of motion for deformable bodies can be considered either in terms of the modal or the nodal coordinates.

Another interesting feature of these equations is that the process of solving the equations of motion for unknown accelerations can be performed either recursively or nonrecursively. It has been shown that for highly serial systems with long chains, a recursive process may yield computational efficiency. Further adaptation of these equations to multiprocessor computers results in a highly efficient simulation package.

8. REFERENCES

1. Wittenburg, J., Dynamics of Systems of Rigid Bodies, B. G. Teubner, 1977.

2. Jerkovsky, W., "The Structure of Multibody Dynamics Equations," J. Guidance and Control, Vol. 1, No. 3, pp. 173-182, May-June 1978.

3. Roberson, R.E., and Schwertassek, R., Dynamics of Multibody Systems, Springer-Verlag, 1988.

4. Kim, S.S., and Vanderploeg, M.J., "A General and Efficient Method for Dynamic Analysis of Mechanical Systems Using Velocity Transformations," ASME J. Mech., Trans., and Auto. in Design, Vol. 108, No. 2, pp. 176-182, June 1986.

5. Ashrafiuon, H., Computer-Aided Optimal Design of Multibody Mechanical Systems Using Symbolic Computation, Ph.D. Dissertation, Mechanical Engineering, SUNY Buffalo, NY, 1988.

6. Nikravesh, P.E., and Gim, G., "Systematic Construction of The Equations of Motion for Multibody Systems Containing Closed Kinematic Loops," Proc., 15th ASME Design Automation Conference, Montreal, Canada, 1989.

7. Nikravesh, P.E., Computer-Aided Analysis of Mechanical Systems, Prentice Hall, 1988.

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

9. Wehage, R.A., "Application of Matrix Partitioning and Recursive Projection to O(n) Solution of Constrint Equations of Motion," Proc., 20th ASME Mechanisms Conference, Orlando, Florida, 1988.

10. Wehage, R.A., "Solution of Multibody Dynamics Using Natural Factors and Iterative Refinement -- Part I: Open Kinematic Loops," Proc., 15th ASME Design Automation Conference, Montreal, Canada, 1989.

11. Wehage, R.A., "Solution of Multibody Dynamics Using Natural Factors and Iterative Refinement -- Part II: Closed Kinematic Loops," Proc., 15th ASME Design Automation Conference, Montreal, Canada, 1989.

12. Rein, U., Implementation of a Recursive Projection Algorithm in a General Purpose Computer Program, Joint Diploma Thesis, Mechanical Engineering, Univ. of Arizona and Univ. of Stuttgart, 1989.

13. Song, J.O., and Haug, E.J., "Dynamic Analysis of Planar Flexible Mechanisms," Computer Methods in Applied Mechanics and Engineering, Vol. 24, pp. 359-381, 1980.

14. Shabana, A., and Wehage, R.A., "Variable Degree of Freedom Component Mode Analysis of Inertia Variant Flexible Mechanical Systems'" ASME J. of Mech., Trans., and Auto. in Design, Vol. 105, pp. 371-378, 1983.

15. Sunada, W., and Dubowsky, S., "The Application of Finite Element Method to the Dynamic Analysis of Flexible Spatial and Co-Planar Linkage Systems," ASME J. of Mech. Design, Vol. 103, pp. 643-651, 1981.