Recursive Flexible Multibody System Dynamics Using Spatial Operators

A. Jain and G. Rodriguez
Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

This paper uses spatial operators to develop new spatially recursive dynamics algorithms for flexible multibody systems. The operator description of the dynamics is identical to that for rigid multibody systems. Assumed-mode models are used for the deformation of each individual body. The algorithms are based on two spatial operator factorizations of the system mass matrix. The first (Newton-Euler) factorization of the mass matrix leads to recursive algorithms for the inverse dynamics, mass matrix evaluation, and composite-body forward dynamics for the system. The second (innovations) factorization of the mass matrix, leads to an operator expression for the mass matrix inverse and to a recursive articulated-body forward dynamics algorithm. The primary focus is on serial chains, but extensions to general topologies are also described. A comparison of computational costs shows that the articulated-body, forward dynamics algorithm is much more efficient than the composite-body algorithm for most flexible multibody systems.

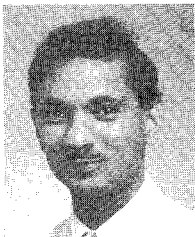## Nomenclature*

General Quantities

$\text{col}\{x(k)\}$ = column vector whose kth element is $x(k)$

$\text{diag}\{x(k)\}$ = block diagonal matrix whose kth diagonal element is $x(k)$

$l(x,y)$ = vector from point/frame x to point/frame y, $\in \mathfrak{R}^3$

$\tilde{x}$ = $[x]^\times \in \mathfrak{R}^{3\times3}$; skew-symmetric cross-product matrix associated with the three-dimensional vector x

$\dot{x}$ = dx/dt; time derivative of x with respect to an inertial frame

$\overset{\circ}{x}$ = time derivative of x with respect to the body-fixed (rotating) frame

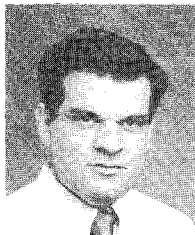$\phi(x,y) = \begin{bmatrix} I & \tilde{l}(x,y) \\ 0 & I \end{bmatrix} \in \mathfrak{R}^{6\times6}$; spatial transformation operator that transforms spatial velocities and forces between points/frames x and y

Individual Body Nodal Data

$\mathfrak{F}_k$ = body reference frame with respect to which the deformation field for the kth body is measured; the motion of this frame characterizes the motion of the kth body as a rigid body

j_k = jth node on the kth body

$K_s(k)$ = structural stiffness matrix for the kth body, $\in \mathfrak{R}^{6n_s(k)\times 6n_s(k)}$

Abhinandan Jain received his Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute. He has been at the Jet Propulsion Laboratory since 1988. His research interests are in the areas of nonlinear systems, robot modeling and control, and control of flexible structures. At JPL he has been conducting research on the dynamic modeling and control of multiple robot systems cooperatively manipulating task objects. He has also been involved in the development of efficient algorithms for the dynamics, kinematics, and control of multibody systems.

Guillermo Rodriguez received the Ph.D. degree in control theory from the University of California at Los Angeles in 1974. He has been with the Jet Propulsion Laboratory, California Institute of Technology, since graduating. He has participated in the development of on-board guidance and control systems for several planetary spacecraft. He has also been the supervisor of technical groups conducting research in the control of large flexible space systems and in applications of machine intelligence to robotics. His research interests include estimation theory, dynamics and control of multibody systems, and control architectures for autonomous robots.

*Coordinate-free spatial notation[1,2] is used in this paper. A *spatial velocity* of a frame is a six-dimensional quantity whose upper three elements are the angular velocity and whose lower three elements are the linear velocity. A *spatial force* is a six-dimensional quantity whose upper three elements are a moment vector and whose lower three elements are a force vector.

A variety of indices are used to identify different spatial quantities. Some examples are as follows: $V_s(j_k)$, the spatial velocity of the jth node on the kth body; $V_s(k) = \text{col}\{V_s(j_k)\}$, the composite vector of spatial velocities of all the nodes on the kth body; $V_s = \text{col}\{V_s(k)\}$, the vector of spatial velocities of all the nodes for all the bodies in the serial chain. The index k is used to refer to both the kth body as well as the kth body reference frame $\mathfrak{F}_k$, with the usage being apparent from the context. Some key quantities are defined in the Nomenclature (see also Fig. 1).

$l(k,j_k)$ = vector from $\mathfrak{F}_k$ to the location (after deformation) of the jth node reference frame on the kth body, $l_0(k,j_k) + \delta_l(j_k) \in \mathfrak{R}^3$

$l_0(k,j_k)$ = vector from $\mathfrak{F}_k$ to the location (before deformation) of the jth node reference frame on the kth body, $\in \mathfrak{R}^3$

$M_s(j_k)$ $= \begin{bmatrix} \mathfrak{J}(j_k) & m(j_k)\tilde{p}(j_k) \\ -m(j_k)\tilde{p}(j_k) & m(j_k)I \end{bmatrix} \in \mathfrak{R}^{6 \times 6};$ spatial inertia about the nodal reference frame for the jth node on the kth body

$M_s(k)$ = structural mass matrix for the kth body, $\mathrm{diag}\{M_s(j_k)\} \in \mathfrak{R}^{6ns(k) \times 6ns(k)}$

$m(j_k)$ = mass of the jth node on the kth body

$n_s(k)$ = number of nodes on the kth body

$p(j_k)$ = vector from the nodal reference frame to the node center of mass for the jth node on the kth body, $\in \mathfrak{R}^3$

$\mathfrak{J}(j_k)$ = inertia tensor about the nodal reference frame for the jth node on the kth body, $\in \mathfrak{R}^{3 \times 3}$

$u(j_k)$ = spatial displacement of node j_k; the translational component of $u(j_k)$ is $\delta_l(j_k)$, whereas its time derivative with respect to the body frame $\mathfrak{F}_k$ is

$$\mathring{u}(j_k) = \begin{bmatrix} \delta_\omega(j_k) \\ \delta_v(j_k) \end{bmatrix}$$

$\delta_l(j_k)$ = translational deformation of the jth node on the kth body, $\in \mathfrak{R}^3$

$\delta_v(j_k)$ = deformation linear velocity of the jth node on the kth body with respect to the body frame $\mathfrak{F}_k$, $\in \mathfrak{R}^3$

$\delta_\omega(j_k)$ = deformation angular velocity of the jth node on the kth body with respect to the body frame $\mathfrak{F}_k$, $\in \mathfrak{R}^3$

Individual Body Modal Data

$K_m(k)$ = modal stiffness matrix for the kth body, $\in \mathfrak{R}^{\mathfrak{N}(k) \times \mathfrak{N}(k)}$

$M_m(k)$ = modal mass matrix for the kth body, $\in \mathfrak{R}^{\mathfrak{N}(k) \times \mathfrak{N}(k)}$

$\mathfrak{N}(k)$ $= n_m(k) + 6$; number of deformation plus rigid-body degrees of freedom for the kth body

$n_m(k)$ = number of assumed modes for the kth body

$\eta(k)$ = vector of modal deformation variables for the kth body, $\in \mathfrak{R}^{nm(k)}$

$\Pi(k)$ $= \mathrm{col}\{\Pi^j(k)\} \in \mathfrak{R}^{6ns(k) \times 6nm(k)};$ the modal matrix for the kth body [The rth column of $\Pi(k)$ is denoted $\Pi_r(k) \in \mathfrak{R}^{6ns(k)}$ and is the mode shape function for the rth assumed mode for the kth body; the deformation field for the kth body is given by $u(k) = \Pi(k)\eta(k)$, whereas $\mathring{u}(k) = \Pi(k)\mathring{\eta}(k)$.]

$\Pi^j(k)$ $= [\Pi^j_1(k), \dots, \Pi^j_{n_m(k)}] \in \mathfrak{R}^{6 \times nm(k)};$ modal spatial influence vector for the jth node [The spatial deformation of node j_k is given by $u(j_k) = \Pi^j(k)\eta(k)$.]

$\Pi^j_r(k)$ = modal spatial displacement vector for the rth mode at the j_kth nodal reference frame, $\in \mathfrak{R}^6$

Multibody Data

$\mathfrak{C}(k)$ $= \begin{bmatrix} [\Pi^t(k)]^* \\ \phi(k,t_k) \end{bmatrix} \in \mathfrak{R}^{\mathfrak{N}(k) \times 6};$ relates spatial forces and velocities between node t_k and frame $\mathfrak{F}_k$

$a_m(k)$ = modal Coriolis and centrifugal accelerations for the kth body, $\in \mathfrak{R}^{\mathfrak{N}(k)}$

$B(k)$ $= [\phi(k,1_k), \phi(k,2_k), \dots, \phi(k,n_s(k)] \in \mathfrak{R}^{6 \times 6ns(k)};$ relates the spatial velocity of frame $\mathfrak{F}_k$ to the spatial velocities of all of the nodes on the

kth body when the body is regarded as being rigid

$\mathfrak{B}(k+1,k)$ $= [0, \phi(t_{k+1}, k)] \in \mathfrak{R}^{6 \times \mathfrak{N}(k)};$ relates spatial forces and velocities between node t_{k+1} and frame $\mathfrak{F}_k$

$b_m(k)$ = modal gyroscopic forces for the kth body, $\in \mathfrak{R}^{\mathfrak{N}(k)}$

$C(k,k-1)$ $= \begin{bmatrix} 0 \\ \vdots \\ \phi(t_k, k-1) \\ \vdots \\ 0 \end{bmatrix} \in \mathfrak{R}^{6ns(k) \times 6}$

$\mathfrak{C}$ = vector of Coriolis, centrifugal, and elastic forces for the multibody system, $\in \mathfrak{R}^{\mathfrak{N}}$

d_k = node on the kth body to which the kth hinge is attached

$f(k)$ = effective spatial force at frame $\mathfrak{F}_k$, $\in \mathfrak{R}^6$

$f_m(k)$ = modal spatial force of interaction between the kth and $(k+1)$th bodies, $\in \mathfrak{R}^{\mathfrak{N}(k)}$

$f_s(j_k)$ = spatial force at node j_k, $\in \mathfrak{R}^6$

$H_{\mathfrak{F}}(k)$ $= H(k)\phi(\mathfrak{O}_k, k) \in \mathfrak{R}^{nr(k) \times 6};$ joint map matrix referred to frame $\mathfrak{F}_k$ for the kth hinge

$H^*(k)$ = joint map matrix for the kth hinge; $\in \mathfrak{R}^{6 \times nr(k)}$ [We have that $\Delta_V(k) = H^*(k)\beta(k)$.]

$\mathfrak{H}(k)$ $= \begin{bmatrix} I & -[\Pi^d_{\mathfrak{F}}(k)]^* \\ 0 & H_{\mathfrak{F}}(k) \end{bmatrix} \in \mathfrak{R}^{\mathfrak{N}(k) \times \mathfrak{N}(k)};$ (deformation plus hinge) modal joint map matrix for the kth body

$\mathfrak{M}$ = multibody system mass matrix, $\in \mathfrak{R}^{\mathfrak{N} \times \mathfrak{N}}$

N = number of bodies in the serial flexible multibody system

$\mathfrak{N}$ $= \Sigma_{k=1}^N \mathfrak{N}(k);$ overall deformation plus hinge degrees of freedom for the serial chain

$\mathfrak{N}(k)$ $= n_m(k) + n_r(k);$ number of deformation plus hinge degrees of freedom for the kth body

$\overline{\mathfrak{N}}$ $= \Sigma_{k=1}^N \overline{\mathfrak{N}}(k);$ overall degrees of freedom in the serial chain obtained by disregarding the hinge constraints

$n_r(k)$ = number of degrees of freedom for the kth hinge

$\mathfrak{O}_k$ = reference frame for the kth hinge on the kth body; this frame is fixed to node d_k

$\mathfrak{O}_k^+$ = reference frame for the kth hinge on the $(k+1)$th body; this frame is fixed to node t_{k+1}

$T(k)$ = generalized force for the kth body, $\in \mathfrak{R}^{\mathfrak{N}(k)}$

t_k = node on the kth body to which the $(k-1)$th hinge is attached

$V(k)$ $= V(\mathfrak{F}_k) = \begin{bmatrix} \omega(k) \\ v(k) \end{bmatrix} \in \mathfrak{R}^6;$ spatial velocity of the kth body reference frame $\mathfrak{F}_k$, with $\omega(k)$ and $v(k)$ denoting the angular and linear velocities respectively of frame $\mathfrak{F}_k$

$V(\mathfrak{O}_k)$ = spatial velocity of frame $\mathfrak{O}_k$, $\in \mathfrak{R}^6$

$V(\mathfrak{O}_k^+)$ = spatial velocity of frame $\mathfrak{O}_k^+$, $\in \mathfrak{R}^6$

$V_m(k)$ $= \begin{bmatrix} \mathring{\eta}(k) \\ V(k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)};$ modal spatial velocity of the kth body

$V_s(j_k)$ = spatial velocity of the jth node on the kth body, $\in \mathfrak{R}^6$

$\alpha_m(k)$ $= \mathring{V}_m(k) \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)};$ modal spatial acceleration of the kth body

$\alpha_s(j_k)$ = spatial acceleration of the jth node on the kth body, $\in \mathfrak{R}^6$

$\beta(k)$ = vector of generalized velocities for the kth hinge, $\in \mathfrak{R}^{nr(k)}$

$$\Delta_V(k) = \begin{bmatrix} \Delta_\omega(k) \\ \Delta_v(k) \end{bmatrix} \in \mathfrak{R}^6;$$

relative spatial velocity for the kth hinge defined as the spatial velocity of frame $\mathcal{O}_k$ with respect to frame $\mathcal{O}_k^+$

$\theta(k)$ = vector of configuration variables for the kth hinge, $\in \mathfrak{R}^{n_r(k)}$

$$\Phi(k+1,k) = \mathcal{Q}(k+1)\mathcal{B}(k+1,k)$$
$$= \begin{bmatrix} 0 & [\Pi'(k+1)]^*\phi(t_{k+1},k) \\ 0 & \phi(k+1,k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k+1) \times \overline{\mathfrak{N}}(k)};$$

interbody transformation operator that relates modal spatial forces and velocities between the kth and $(k+1)$th bodies

$$\chi(k) = \begin{bmatrix} \dot{\eta}(k) \\ \beta(k) \end{bmatrix} \in \mathfrak{R}^{\mathfrak{N}(k)};$$

vector of (deformation plus hinge) generalized velocities for the kth body

$$\vartheta(k) = \begin{bmatrix} \eta(k) \\ \theta(k) \end{bmatrix} \in \mathfrak{R}^{\mathfrak{N}(k)};$$

vector of (deformation plus hinge) generalized configuration variables for the kth body

I. Introduction

THIS paper uses spatial operators[1,2] to formulate dynamics and develop efficient recursive algorithms for flexible multibody systems. Flexible spacecraft, limber space manipulators, and vehicles are important examples of flexible multibody systems. Key features of these systems are the large number of degrees of freedom and the complexity of their dynamics models.

The main contributions of the paper are 1) providing a high-level architectural understanding of the structure of the mass matrix and its inverse; 2) showing that the high-level expressions can be easily implemented within the very well understood Kalman filtering and smoothing architecture; 3) developing very efficient inverse and forward dynamics recursive algorithms; and 4) analyzing the computational cost of the new algorithms. These contributions add to the rapidly developing body of research in the recursive dynamics of flexible multibody systems.[3-5]

It is assumed that the bodies undergo small deformations so that a linear model for elasticity can be used. However, large articulation at the hinges is allowed. No special assumptions are made regarding the geometry of the component bodies. To maximize applicability, the algorithms developed here use finite element and/or assumed-mode models for body flexibility. For notational simplicity, and without any loss in generality, the main focus of this paper is on flexible multibody serial chains. Extensions to tree and closed-chain topologies are discussed.

In Sec. II we derive the equations of motion and recursive relationships for the modal velocities, modal accelerations, and modal forces. This section also contains a derivation of the Newton-Euler operator factorization of the system mass matrix. A recursive Newton-Euler inverse dynamics algorithm to compute the vector of generalized forces corresponding to a given state and vector of generalized accelerations is described in Sec. III.

In Sec. IV, the Newton-Euler factorization of the mass matrix is used to develop a partly recursive composite-body forward dynamics algorithm for computing the generalized accelerations of the system. The recursive part is for computing the multibody system mass matrix. This forward dynamics algorithm is in the vein of well-established approaches[6,7] that require the explicit computation and inversion of the system mass matrix. However, the new algorithm is more efficient because the mass matrix is computed recursively and because the detailed recursive computations follow the high-

level architecture (i.e., roadmap) provided by the Newton-Euler factorization.

In Sec. V we derive new operator factorization and inversion results for the mass matrix that lead to the recursive articulated-body forward dynamics algorithm. A new mass matrix operator factorization, referred to as the "innovations" factorization, is developed. The individual factors in the innovations factorization are square and invertible operators. This is in contrast to the Newton-Euler factorization in which the factors are not square and therefore not invertible. The innovations factorization leads to an operator expression for the inverse of the mass matrix. Based on this expression, in Sec. VI we develop the recursive articulated-body forward dynamics algorithm for the multibody system. This algorithm is an alternative to the composite-body forward dynamics algorithm and requires neither the explicit formation of the system mass matrix nor its inversion. The structure of this recursive algorithm closely resembles those found in the domain of Kalman filtering and smoothing.[8]

In Sec. VII we compare the computational costs for the two forward dynamics algorithms. It is shown that the articulated-body forward dynamics algorithm is much more efficient than the composite-body forward dynamics algorithm for typical flexible multibody systems. In Sec. VIII we discuss the extensions of the formulation and algorithms in this paper to tree and closed-chain topology multibody systems.

II. Equations of Motion for Flexible Serial Chains

In this section, we develop the equations of motion for a serial, flexible, multibody system with N flexible bodies. Each flexible body is assumed to have a lumped mass model consisting of a collection of nodal rigid bodies. Such models are typically developed using standard finite element structural analysis software. The number of nodes on the kth body is denoted $n_s(k)$. The jth node on the kth body is referred to as the j_kth node. Each body has associated with it a body reference frame, denoted $\mathfrak{F}_k$ for the kth body. The deformations of the nodes on the body are described with respect to this body reference frame, whereas the rigid body motion of the kth body is characterized by the motion of frame $\mathfrak{F}_k$.

The six-dimensional spatial deformation (slope plus translational) of node j_k (with respect to frame $\mathfrak{F}_k$) is denoted $u(j_k) \in \mathfrak{R}^6$. The overall deformation field for the kth body is defined as the vector

$$u(k) = \text{col}\{u(j_k)\} \in \mathfrak{R}^{6n_s(k)}$$

The vector from frame $\mathfrak{F}_k$ to the reference frame on node j_k is denoted $l(k,j_k) \in \mathfrak{R}^3$.

With $M_s(j_k) \in \mathfrak{R}^{6 \times 6}$ denoting the spatial inertia of the jth node, the structural mass matrix for the kth body $M_s(k)$ is the block diagonal matrix

$$\text{diag}\{M_s(j_k)\} \in \mathfrak{R}^{6n_s(k) \times 6n_s(k)}$$

The structural stiffness matrix is denoted

$$K_s(k) \in \mathfrak{R}^{6n_s(k) \times 6n_s(k)}$$

Both $M_s(k)$ and $K_s(k)$ are typically generated using finite element analysis.

As shown in Fig. 1, the bodies in the serial chain are numbered in increasing order from tip to base. We use the terminology inboard (outboard) to denote the direction along the serial chain toward (away from) the base body. The kth body is attached on the inboard side to the $(k+1)$th body via the kth hinge and on the outboard side to the $(k-1)$th body via the $(k-1)$th hinge. On the kth body, the node to which the outboard hinge [the $(k-1)$th hinge] is attached is referred to as node t_k, whereas the node to which the inboard hinge (the kth hinge) is attached is denoted node d_k. Thus the kth
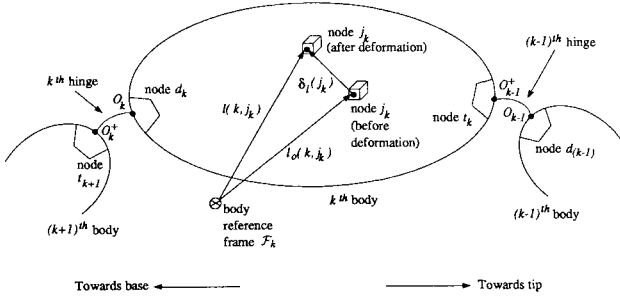
Fig. 1   Links and hinges in a flexible serial multibody system.

hinge couples together nodes d_k and t_{k+1}. Attached to each of these pairs of adjoining nodes are the kth hinge reference frames denoted Θ_k and Θ_k^+, respectively. The number of degrees of freedom for the kth hinge is denoted $n_r(k)$. The vector of configuration variables for the kth hinge is denoted $\theta(k) \in \Re^{n_r(k)}$, whereas its vector of generalized speeds is denoted $\beta(k) \in \Re^{n_r(k)}$. In general, when there are nonholonomic hinge constraints, the dimensionality of $\beta(k)$ may be less than that of $\theta(k)$. For notational convenience, and without any loss in generality, it is assumed here that the dimensions of the vectors $\theta(k)$ and $\beta(k)$ are equal. In most situations, $\beta(k)$ is simply $\dot{\theta}$. However, there are many cases where the use of quasicoordinates simplifies the dynamical equations of motion and an alternative choice for $\beta(k)$ may be preferable. The relative spatial velocity $\Delta_V(k)$ across the hinge is given by $H^*(k)\beta(k)$, where $H^*(k)$ denotes the joint map matrix for the kth hinge.

Assumed modes typically are used to represent the deformation of flexible bodies, and there is a large body of literature dealing with their proper selection. There is however a close relationship between the choice of a body reference frame and the type of assumed modes. The complete motion of the flexible body is contained in the knowledge of the motion of the body reference frame and the deformation of the body as seen from this body frame. In the multibody context, it is often convenient to choose the location of the kth body reference frame $\mathcal{F}_k$ as a material point on the body and fixed to node d_k at the inboard hinge. For this choice, the assumed modes are cantilever modes and node d_k exhibits zero deformation $[u(d_k)=0]$. Free-free modes are also used to represent body deformation and are often preferred for control analysis and design. For these modes, the reference frame $\mathcal{F}_k$ is not fixed to any node but is rather assumed to be fixed to the undeformed body, and as a result all nodes exhibit nonzero deformation. The dynamics modeling and algorithms developed here handle both types of modes, with some additional computational simplifications arising from Eq. (1) when cantilever modes are used. For a related discussion regarding the choice of reference frame and modal representations for a flexible body see Ref. 9.

We assume here that a set of $n_m(k)$ assumed modes has been chosen for the kth body. Let $\Pi_r^j(k) \in \Re^6$ denote the modal spatial displacement vector at the j_kth node for the rth mode. The modal spatial displacement influence vector $\Pi^j(k) \in \Re^{6 \times n_m(k)}$ for the j_kth node and the modal matrix $\Pi(k) \in \Re^{6n_s(k) \times n_m(k)}$ for the kth body are defined as follows:

$$\Pi^j(k) = \left[\Pi_1^j(k), \ldots, \Pi_{n_m(k)}^j(k)\right] \quad \text{and} \quad \Pi(k) = \text{col}\{\Pi^j(k)\}$$

The rth column of $\Pi(k)$ is denoted $\Pi_r(k)$ and defines the mode shape for the rth assumed mode for the kth body. Note that for cantilever modes we have

$$\Pi_r^d(k) = 0 \quad \text{for} \quad r = 1 \cdots n_m(k) \tag{1}$$

With $\eta(k) \in \Re^{n_m(k)}$ denoting the vector of modal deformation variables for the kth body, the spatial deformation of node

j_k and the spatial deformation field $u(k)$ for the kth body are given by

$$u(j_k) = \Pi^j(k)\eta(k) \quad \text{and} \quad u(k) = \Pi(k)\eta(k) \tag{2}$$

The vector of generalized configuration variables $\vartheta(k)$ and generalized speeds $\chi(k)$ for the kth body are defined as

$$\vartheta(k) \triangleq \begin{bmatrix} \eta(k) \\ \theta(k) \end{bmatrix} \in \Re^{\Re(k)}$$

$$\chi(k) \triangleq \begin{bmatrix} \dot{\eta}(k) \\ \beta(k) \end{bmatrix} \in \Re^{\Re(k)} \tag{3}$$

where $\Re(k) \triangleq n_m(k) + n_r(k)$. The overall vectors of generalized configuration variables ϑ and generalized speeds χ for the serial multibody system are given by

$$\vartheta \triangleq \text{col}\{\vartheta(k)\} \in \Re^{\Re} \quad \text{and} \quad \chi \triangleq \text{col}\{\chi(k)\} \in \Re^{\Re} \tag{4}$$

where

$$\Re \triangleq \sum_{k=1}^{N} \Re(k)$$

denotes the overall number of degrees of freedom for the multibody system. The *state* of the multibody system is defined by the pair of vectors $\{\vartheta, \chi\}$. For a given system state $\{\vartheta, \chi\}$, the equations of motion define the relationship between the vector of generalized accelerations $\dot{\chi}$ and the vector of generalized forces $T \in \Re^{\Re}$ for the system. The inverse dynamics problem consists of computing the vector of generalized forces T for a prescribed set of generalized accelerations $\dot{\chi}$. The forward dynamics problem is the converse one and consists of computing the set of generalized accelerations $\dot{\chi}$ resulting from a set of generalized forces T. The equations of motion for the system are developed in the remainder of this section.

A. Recursive Propagation of Velocities

Let $V(k) \in \Re^6$ denote the spatial velocity of the kth body reference frame $\mathcal{F}_k$. The spatial velocity $V_s(t_{k+1}) \in \Re^6$ of node t_{k+1} (on the inboard of the kth hinge) is related to the spatial velocity $V(k+1)$ of the $(k+1)$th body reference frame $\mathcal{F}_{k+1}$, and the modal deformation variable rates $\dot{\eta}(k+1)$ as follows:

$$V_s(t_{k+1}) = \phi^*(k+1, t_{k+1})V(k+1) + \mathring{u}(t_{k+1})$$

$$= \phi^*(k+1, t_{k+1})V(k+1) + \Pi^t(k+1)\dot{\eta}(k+1) \tag{5}$$

The aforementioned spatial transformation operator $\phi(x, y) \in \Re^{6 \times 6}$ is defined to be

$$\phi(x, y) = \begin{bmatrix} I & \tilde{l}(x, y) \\ 0 & I \end{bmatrix} \tag{6}$$

where $l(x, y) \in \Re^3$ denotes the vector between the points x and y. Note that the following important (group) property holds for arbitrary points x, y, and z:

$$\phi(x, y)\phi(y, z) = \phi(x, z)$$

As in Eq. (5), and all through this paper, the index k will be used to refer to both the kth body as well as to the kth body reference frame $\mathcal{F}_k$ with the specific usage being evident from the context. Thus, for instance, $V(k)$ and $\phi(k, t_k)$ are the same as $V(\mathcal{F}_k)$ and $\phi(\mathcal{F}_k, t_k)$, respectively.

The spatial velocity $V(\Theta_k^+)$ of frame Θ_k^+ (on the inboard side of the kth hinge) is related to $V_s(t_{k+1})$ via

$$V(\Theta_k^+) = \phi^*(t_{k+1}, \Theta_k)V_s(t_{k+1}) \tag{7}$$

Since the relative spatial velocity $\Delta_V(k)$ across the kth hinge is given by $H^*(k)\beta(k)$, the spatial velocity $V(\mathcal{O}_k)$ of frame $\mathcal{O}_k$ on the outboard side of the kth hinge is

$$V(\mathcal{O}_k) = V(\mathcal{O}_k^+) + H^*(k)\beta(k) \tag{8}$$

The spatial velocity $V(k)$ of the kth body reference frame is given by

$$V(k) = \phi^*(\mathcal{O}_k,k)\left[V(\mathcal{O}_k) - \hat{u}(d_k)\right]$$
$$= \phi^*(\mathcal{O}_k,k)\left[V(\mathcal{O}_k) - \Pi^d(k)\dot{\eta}(k)\right] \tag{9}$$

Putting together Eqs. (5) and (7-9), it follows that

$$V(k) = \phi^*(k+1,k)V(k+1) + \phi^*(t_{k+1},k)\Pi^t(k+1)\dot{\eta}(k+1)$$
$$+ \phi^*(\mathcal{O}_k,k)\left[H^*(k)\beta(k) - \Pi^d(k)\dot{\eta}(k)\right] \tag{10}$$

Thus with $\overline{\mathfrak{N}}(k) \triangleq n_m(k) + 6$, and using Eq. (10), the modal spatial velocity $V_m(k) \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)}$ for the kth body is given by

$$V_m(k) \triangleq \begin{bmatrix} \dot{\eta}(k) \\ V(k) \end{bmatrix} = \Phi^*(k+1,k)V_m(k+1)$$
$$+ \mathcal{K}^*(k)\chi(k) \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)} \tag{11}$$

where the interbody transformation operator $\Phi(.,.)$ and the modal joint map matrix $\mathcal{K}(k)$ are defined as

$$\Phi(k+1,k) \triangleq \begin{bmatrix} 0 & [\Pi^t(k+1)]^*\phi(t_{k+1},k) \\ 0 & \phi(k+1,k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k+1)\times\overline{\mathfrak{N}}(k)} \tag{12}$$

$$\mathcal{K}(k) \triangleq \begin{bmatrix} I & [\Pi^d_{\mathfrak{F}}(k)]^* \\ 0 & H_{\mathfrak{F}}(k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)\times\overline{\mathfrak{N}}(k)} \tag{13}$$

where

$$H_{\mathfrak{F}}(k) \triangleq H(k)\phi(\mathcal{O}_k,k) \in \mathfrak{R}^{n_r(k)\times 6}$$

$$\Pi^d_{\mathfrak{F}}(k) \triangleq \phi^*(\mathcal{O}_k,k)\Pi^d(k) \in \mathfrak{R}^{6\times\overline{\mathfrak{N}}(k)}$$

Note that

$$\Phi(k+1,k) = \mathcal{C}(k+1)\mathcal{B}(k+1,k) \tag{14}$$

where

$$\mathcal{C}(k) \triangleq \begin{bmatrix} [\Pi^t(k)]^* \\ \phi(k,t_k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)\times 6}$$

$$B(k+1,k) \triangleq \begin{bmatrix} 0, & \phi(t_{k+1},k) \end{bmatrix} \in \mathfrak{R}^{6\times\overline{\mathfrak{N}}(k)} \tag{15}$$

Also, the modal joint map matrix $\mathcal{K}(k)$ can be partitioned as

$$\mathcal{K}(k) = \begin{bmatrix} \mathcal{K}_f(k) \\ \mathcal{K}_r(k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)\times\overline{\mathfrak{N}}(k)} \tag{16}$$

where

$$\mathcal{K}_f(k) \triangleq \begin{bmatrix} I, & -[\Pi^d_{\mathfrak{F}}(k)]^* \end{bmatrix} \in \mathfrak{R}^{n_m(k)\times\overline{\mathfrak{N}}(k)}$$

$$\mathcal{K}_r(k) \triangleq \begin{bmatrix} 0, & H_{\mathfrak{F}}(k)\phi(\mathcal{O}_k,k) \end{bmatrix} \in \mathfrak{R}^{n_r(k)\times\overline{\mathfrak{N}}(k)} \tag{17}$$

With

$$\overline{\mathfrak{N}} = \sum_{k=1}^{N} \overline{\mathfrak{N}}(k)$$

we define the spatial operator $\mathcal{E}_\Phi$ as

$$\mathcal{E}_\Phi \triangleq \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \Phi(2,1) & 0 & \cdots & 0 & 0 \\ 0 & \Phi(3,2) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \Phi(N,N-1) & 0 \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}\times\overline{\mathfrak{N}}} \tag{18}$$

Using the fact that $\mathcal{E}_\Phi$ is nilpotent (i.e., $\mathcal{E}_\Phi^N = 0$), we define the spatial operator Φ as

$$\Phi \triangleq [I - \mathcal{E}_\Phi]^{-1} = I + \mathcal{E}_\Phi + \cdots + \mathcal{E}_\Phi^{N-1}$$
$$= \begin{bmatrix} I & 0 & \cdots & 0 \\ \Phi(2,1) & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(N,1) & \Phi(N,2) & \cdots & I \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}\times\overline{\mathfrak{N}}} \tag{19}$$

where

$$\Phi(i,j) \triangleq \Phi(i,i-1)\cdots\Phi(j+1,j) \qquad \text{for} \qquad i > j$$

Also, define the spatial operator

$$\mathcal{K} \triangleq \text{diag}\{\mathcal{K}(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}}\times\overline{\mathfrak{N}}}$$

When these spatial operators are used, and with

$$V_m \triangleq \text{col}\{V_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}}}$$

defined, it follows from Eq. (11) that the spatial operator expression for V_m is given by

$$V_m = \Phi^*\mathcal{K}^*\chi \tag{20}$$

B. Modal Mass Matrix for a Single Body

With $V_s(j_k) \in \mathfrak{R}^6$ denoting the spatial velocity of node j_k, and

$$V_s(k) \triangleq \text{col}\{V_s(j_k)\} \in \mathfrak{R}^{6n_s(k)}$$

the vector of all nodal spatial velocities for the kth body, it follows [see Eq. (5)] that

$$V_s(k) = B^*(k)V(k) + \hat{u}(k) = \begin{bmatrix} \Pi(k), & B^*(k) \end{bmatrix}V_m(k) \tag{21}$$

where

$$B(k) \triangleq \begin{bmatrix} \phi(k+1_k), \phi(k,2_k), \ldots, \phi(k,n_s(k)) \end{bmatrix} \in \mathfrak{R}^{6\times 6n_s(k)} \tag{22}$$

Since $M_s(k)$ is the structural mass matrix of the kth body, when Eq. (21) is used, the kinetic energy of the kth body can be written in the form

$$\tfrac{1}{2} V_s^*(k)M_s(k)V_s(k) = \tfrac{1}{2} V_m^*(k)M_m(k)V_m(k)$$

where

$$M_m(k) \triangleq \begin{bmatrix} \Pi^*(k) \\ B(k) \end{bmatrix}M_s(k)\begin{bmatrix} \Pi(k), & B^*(k) \end{bmatrix}$$
$$= \begin{bmatrix} \Pi^*(k)M_s(k)\Pi(k) & \Pi^*(k)M_s(k)B^*(k) \\ B(k)M_s(k)\Pi(k) & B(k)M_s(k)B^*(k) \end{bmatrix}$$
$$= \begin{bmatrix} M_m^{ff}(k) & M_m^{fr}(k) \\ M_m^{rf}(k) & M_m^{rr}(k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)\times\overline{\mathfrak{N}}(k)} \tag{23}$$

Corresponding to the generalized speeds vector $\chi(k)$, $M_m(k)$ as defined in Eq. (23) is the modal mass matrix of the kth body. In the block partitioning in Eq. (23), the superscripts f and r denote the flexible and rigid blocks, respectively. Thus, $M_m^{ff}(k)$ represents the flex/flex coupling block, while $M_m^{fr}(k)$ is the flex/rigid coupling block of $M_m(k)$. We will use this notational convention throughout this paper. Note that $M_m^{rr}(k)$ is precisely the rigid body spatial inertia of the kth body. Indeed, $M_m(k)$ reduces to the rigid-body spatial inertia when the body flexibility is ignored, i.e., no modes are used, since in this case $n_m(k) = 0$ [and $\Pi(k)$ is null].

Because the vector $l(k, j_k)$ from $\mathcal{F}_k$ to node j_k depends on the deformation of the node, the operator $B(k)$ is also deformation dependent. From Eq. (23) it follows that while the block $\mathfrak{M}_m^{ff}(k)$ is deformation independent, both the blocks $M_m^{fr}(k)$ and M_m^{rr} are deformation dependent. The detailed expression for the modal mass matrix can be defined using modal integrals that are computed as a part of the finite element structural analysis of the flexible bodies. These expressions for the modal integrals and the modal mass matrix of the kth body can be found in Ref. 10. Often the deformation dependent parts of the modal mass matrix are ignored, and free-free eigenmodes are used for the assumed modes $\Pi(k)$. When this is the case, $M_m^{fr}(k)$ is zero and $M_m^{ff}(k)$ is diagonal.

C. Recursive Propagation of Accelerations

Differentiating the velocity recursion equation (11), we obtain the following recursive expression for the modal spatial acceleration $\alpha_m(k) \in \mathfrak{R}^{\mathfrak{N}(k)}$ for the kth body:

$$\alpha_m(k) \triangleq \dot{V}_m(k) = \begin{bmatrix} \ddot{\eta}(k) \\ \alpha(k) \end{bmatrix}$$

$$= \Phi^*(k+1,k)\alpha_m(k+1) + \mathcal{K}^*(k)\dot{\chi}(k) + a_m(k) \quad (24)$$

where $\alpha(k) = \dot{V}(k)$, and the Coriolis and centrifugal acceleration term $a_m(k) \in \mathfrak{R}^{\mathfrak{N}(k)}$ is given by

$$a_m(k) = \frac{d\Phi^*(k+1,k)}{dt} V_m(k+1) + \frac{d\mathcal{K}^*(k)}{dt} \chi(k) \quad (25)$$

The detailed expressions for $a_m(k)$ can be found in Ref. 10. Defining

$$a_m = \mathrm{col}\{a_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}}} \quad \text{and} \quad \alpha_m = \mathrm{col}\{\alpha_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}}}$$

and using spatial operators, we can re-express Eq. (24) in the form

$$\alpha_m = \Phi^*(\mathcal{K}^*\dot{\chi} + a_m) \quad (26)$$

The vector of spatial accelerations of all of the nodes for the kth body

$$\alpha_s(k) \triangleq \mathrm{col}\{\alpha_s(j_k)\} \in \mathfrak{R}^{6n_s(k)}$$

is obtained by differentiating Eq. (21):

$$\alpha_s(k) = \dot{V}_s(k) = \begin{bmatrix} \Pi(k), & B^*(k) \end{bmatrix} \alpha_m(k) + a(k) \quad (27)$$

where

$$a(k) \triangleq \mathrm{col}\{a(j_k)\} = \frac{d\begin{bmatrix} \Pi(k), & B^*(k) \end{bmatrix}}{dt} V_m(k) \in \mathfrak{R}^{6n_s(k)} \quad (28)$$

D. Recursive Propagation of Forces

Let $f(k-1) \in \mathfrak{R}^6$ denote the effective spatial force of interaction, referred to frame $\mathcal{F}_{k-1}$, between the kth and $(k-1)$th bodies across the $(k-1)$th hinge. Recall that the $(k-1)$th hinge is between node t_k on the kth body and node d_{k-1} on the

$(k-1)$th body. With $f_s(j_k) \in \mathfrak{R}^6$ denoting the spatial force at a node j_k, the force balance equation for node t_k is given by

$$f_s(t_k) = \phi(t_k, k-1)f(k-1) + M_s(t_k)\alpha_s(t_k) + b(t_k) + f_K(t_k) \quad (29)$$

For all nodes other than node t_k on the kth body, the force balance equation is of the form

$$f_s(j_k) = M_s(j_k)\alpha_s(j_k) + b(j_k) + f_K(j_k) \quad (30)$$

In Eqs. (29) and (30), $f_K(j_k)$ are components of the vector

$$f_K(k) = K_s(k)u(k) \in \mathfrak{R}^{6n_s(k)}$$

denotes the vector of spatial elastic strain forces for the nodes on the kth body, whereas $b(j_k) \in \mathfrak{R}^6$ denotes the spatial gyroscopic force for node j_k and is given by

$$b(j_k) = \begin{bmatrix} \tilde{\omega}(j_k)\mathcal{J}(j_k)\omega(j_k) \\ m(j_k)\tilde{\omega}(j_k)\tilde{\omega}(j_k)p(j_k) \end{bmatrix} \in \mathfrak{R}^6 \quad (31)$$

where $\omega(j_k) \in \mathfrak{R}^3$ denotes the angular velocity of node j_k. Collecting together the preceding equations and defining

$$C(k, k-1) \triangleq \begin{bmatrix} 0 \\ \vdots \\ \phi(t_k, k-1) \\ \vdots \\ 0 \end{bmatrix} \in \mathfrak{R}^{6n_s(k) \times 6}$$

$$b(k) \triangleq \mathrm{col}\{b(j_k)\} \in \mathfrak{R}^{6n_s(k)} \quad (32)$$

it follows from Eqs. (29) and (30) that

$$f_s(k) = C(k, k-1)f(k-1)$$

$$+ M_s(k)\alpha_s(k) + b(k) + K_s(k)u(k) \quad (33)$$

where

$$f_s(k) \triangleq \mathrm{col}\{f_s(j_k)\} \in \mathfrak{R}^{6n_s(k)}$$

Noting that

$$f(k) = B(k)f_s(k) \quad (34)$$

and using the principle of virtual work, it follows from Eq. (21) that the modal spatial forces $f_m(k) \in \mathfrak{R}^{\mathfrak{N}(k)}$ for the kth body are given by

$$f_m(k) \triangleq \begin{bmatrix} \Pi^*(k) \\ B(k) \end{bmatrix} f_s(k) = \begin{bmatrix} \Pi^*(k)f_s(k) \\ f(k) \end{bmatrix} \quad (35)$$

Premultiplying Eq. (33) by

$$\begin{bmatrix} \Pi^*(k) \\ B(k) \end{bmatrix}$$

and using Eqs. (23), (27), and (35) leads to the following recursive relationship for the modal spatial forces:

$$f_m(k) = \begin{bmatrix} \Pi^*(k)C(k, k-1) \\ B(k)C(k, k-1) \end{bmatrix} f(k-1) + M_m(k)\alpha_m(k)$$

$$+ b_m(k) + K_m(k)\vartheta(k)$$

$$= \begin{bmatrix} [\Pi'(k)]^* \\ \phi(k, t_k) \end{bmatrix} \phi(t_k, k-1)f(k-1) + M_m(k)\alpha_m(k)$$

$$+ b_m(k) + K_m(k)\vartheta(k)$$

$$= \Phi(k, k-1)f_m(k-1) + M_m(k)\alpha_m(k)$$

$$+ b_m(k) + K_m(k)\vartheta(k) \qquad (36)$$

Here we have defined

$$b_m(k) \triangleq \begin{bmatrix} \Pi^*(k) \\ B(k) \end{bmatrix} \left[b(k) + M_s(k)a(k) \right] \in \mathfrak{R}^{\overline{\mathfrak{M}}(k)} \qquad (37)$$

and the modal stiffness matrix

$$K_m(k) \triangleq \begin{bmatrix} \Pi^*(k)K_s(k)\Pi(k) & 0 \\ 0 & 0 \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{M}}(k) \times \overline{\mathfrak{M}}(k)} \qquad (38)$$

The expression for $K_m(k)$ in Eq. (38) uses the fact that the columns of $B^*(k)$ are indeed the deformation dependent rigid body modes for the kth body and hence they do not contribute to its elastic strain energy. Indeed, when a deformation-dependent structural stiffness matrix $K_s(k)$ is used, we have

$$K_s(k)B^*(k) = 0 \qquad (39)$$

However, the common practice (also followed here) of using a constant, deformation-independent structural stiffness matrix leads to the anomalous situation wherein Eq. (39) does not hold exactly. We ignore these fictitious extra terms on the left-hand side of Eq. (39).

The velocity-dependent bias term $b_m(k)$ is formed using modal integrals generated by standard finite element programs, and a detailed expression for it is given in Ref. 10. From Eq. (36), the operator expression for the modal spatial forces

$$f_m \triangleq \mathrm{col}\{f_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{M}}}$$

for all of the bodies in the chain is given by

$$f_m = \Phi(M_m \alpha_m + b_m + K_m \vartheta) \qquad (40)$$

where

$$M_m \triangleq \mathrm{diag}\{M_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{M}} \times \overline{\mathfrak{M}}}$$

$$K_m \triangleq \mathrm{diag}\{K_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{M}} \times \overline{\mathfrak{M}}}$$

$$b_m \triangleq \mathrm{col}\{b_m(k)\} \in \mathfrak{R}^{\overline{\mathfrak{M}}}$$

From the principle of virtual work, the generalized forces vector $T \in \mathfrak{R}^{\mathfrak{M}}$ for the multibody system is given by the expression

$$T = \mathfrak{K}f_m \qquad (41)$$

E. Operator Expression for the System Mass Matrix

Collecting together the operator expressions in Eqs. (20), (26), (40), and (41), we have

$$V_m = \Phi^* \mathfrak{K}^* \chi$$

$$\alpha_m = \Phi^*(\mathfrak{K}^* \dot{\chi} + a_m)$$

$$f_m = \Phi(M_m \alpha_m + b_m + K_m \vartheta)$$

$$= \Phi M_m \Phi^* \mathfrak{K}^* \dot{\chi} + \Phi(M_m \Phi^* a_m + b_m + K_m \vartheta)$$

$$T = \mathfrak{K}f_m = \mathfrak{K}\Phi M_m \Phi^* \mathfrak{K}^* \dot{\chi} + \mathfrak{K}\Phi(M_m \Phi^* a_m + b_m)$$

$$= \mathfrak{M}\dot{\chi} + \mathfrak{C} \qquad (42)$$

where

$$\mathfrak{M} \triangleq \mathfrak{K}\Phi M_m \Phi^* \mathfrak{K}^* \in \mathfrak{R}^{\mathfrak{M} \times \mathfrak{M}}$$

$$\mathfrak{C} \triangleq \mathfrak{K}\Phi(M_m \Phi^* a_m + b_m + K_m \vartheta) \in \mathfrak{R}^{\mathfrak{M}} \qquad (43)$$

Here $\mathfrak{M}$ is the system mass matrix for the serial chain and the expression $\mathfrak{K}\Phi M_m \Phi^* \mathfrak{K}^*$ is referred to as the "Newton-Euler operator factorization" of the mass matrix. $\mathfrak{C}$ is the vector of Coriolis, centrifugal, and elastic forces for the system.

It is noteworthy that the operator expressions for $\mathfrak{M}$ and $\mathfrak{C}$ are identical in form to those for rigid multibody systems.[1,11] Indeed, the similarity is more than superficial, and the key properties of the spatial operators that are used in the analysis and algorithm development for rigid multibody systems also hold for the spatial operators defined here. As a consequence, a large part of the analysis and algorithms for rigid multibody systems can be easily carried over and applied to flexible multibody systems. That is the approach adopted here.

III. Inverse Dynamics Algorithm

This section describes a recursive Newton-Euler inverse dynamics algorithm for computing the generalized forces T, for a given set of generalized accelerations $\dot{\chi}$ and system state $\{\vartheta, \chi\}$. The inverse dynamics algorithm also forms a part of forward dynamics algorithms such as those based on composite body inertias or the conjugate gradient method.[12]

Collecting together the recursive equations in Eqs. (11), (24), (36), and (41), we obtain the following recursive Newton-Euler inverse dynamics algorithm:

$$V_m(N+1) = 0, \qquad \alpha_m(N+1) = 0$$

for $k = N, \dots, 1$

$$V_m(k) = \Phi^*(k+1, k)V_m(k+1) + \mathfrak{K}^*(k)\chi(k)$$

$$\alpha_m(k) = \Phi^*(k+1, k)\alpha_m(k+1) + \mathfrak{K}^*(k)\dot{\chi}(k) + a_m(k)$$

end loop

$$f_m(0) = 0$$

for $k = 1, \dots, N$

$$f_m(k) = \Phi(k, k-1)f_m(k-1) + M_m(k)\alpha_m(k) + b_m(k)$$

$$+ K_m(k)\vartheta(k)$$

$$T(k) = \mathfrak{K}(k)f_m(k)$$

end loop $\qquad (44)$

The structure of this algorithm closely resembles the recursive Newton-Euler inverse dynamics algorithm for rigid multibody systems.[1,13] All external forces on the kth body are handled by absorbing them into the gyroscopic force term $b_m(k)$. Base mobility is handled by attaching an additional six-degree-of-freedom hinge between the mobile base and an inertial frame.

By taking advantage of the special structure of $\Phi(k+1, k)$ and $\mathfrak{K}(k)$ in Eqs. (12) and (13), the Newton-Euler recursions in Eq. (44) can be further simplified. Using block partitioning and the superscripts f and r as before to denote the flexible and rigid components, we have

$$V_m(k) = \begin{bmatrix} V_m^f(k) \\ V_m^r(k) \end{bmatrix}, \qquad \alpha_m(k) = \begin{bmatrix} \alpha_m^f(k) \\ \alpha_m^r(k) \end{bmatrix}$$

$$f_m(k) = \begin{bmatrix} f_m^f(k) \\ f_m^r(k) \end{bmatrix}, \qquad T(k) = \begin{bmatrix} T^f(k) \\ T^r(k) \end{bmatrix}$$

It is easy to verify that Eq. (45) is a simplified version of the inverse dynamics algorithm in Eq. (44).

$$V_m(N+1) = 0, \qquad \alpha_m(N+1) = 0$$

for $k = N, \ldots, 1$

$$V_m^f(k) = \dot{\eta}(k)$$

$$V_m^r(k) = \phi^*(t_{k+1}, k)\mathcal{Q}^*(k+1)V_m(k+1)$$

$$+ \mathcal{HC}_{\mathfrak{F}}^*(k)\beta(k) - \Pi_{\mathfrak{F}}^d(k)\dot{\eta}(k)$$

$$\alpha_m^f(k) = \ddot{\eta}(k)$$

$$\alpha_m^r(k) = \phi^*(t_{k+1}, k)\mathcal{Q}^*(k+1)\alpha_m(k+1)$$

$$+ H_{\mathfrak{F}}^*(k)\dot{\beta}(k) - \Pi_{\mathfrak{F}}^d(k)\ddot{\eta}(k) + a_m^r(k)$$

end loop   (45)

$$f_m(0) = 0$$

for $k = 1, \ldots, N$

$$f_m(k) = \mathcal{Q}(k)\phi(t_k, k-1)f_m^r(k-1) + M_m(k)\alpha_m(k)$$

$$+ b_m(k) + K_m(k)\vartheta(k)$$

$$T(k) = \begin{bmatrix} T^f(k) \\ T^r(k) \end{bmatrix} = \begin{bmatrix} f_m^f(k) - [\Pi_{\mathfrak{F}}^d(k)]^* f_m^r(k) \\ H_{\mathfrak{F}}(k)f_m^r(k) \end{bmatrix}$$

end loop

Flexible multibody systems have actuators typically only at the hinges. Thus, for the kth body, only the subset of the generalized forces vector $T(k)$ corresponding to the hinge actuator forces $T^r(k)$ can be set, whereas the remaining generalized forces $T^f(k)$ are zero. Therefore, in contrast with rigid multibody systems, flexible multibody systems are underactuated systems,[14] since the number of available actuators is less than the number of motion degrees of freedom in the system. For such underactuated systems, the inverse dynamics computations for the generalized force T are meaningful only when the prescribed generalized accelerations $\dot{\chi}$ form a consistent data set. For a consistent set of generalized accelerations, the inverse dynamics computations will lead to a generalized force vector T such that $T^f(.) = 0$.

IV. Composite-Body Forward Dynamics Algorithm

The forward dynamics problem for a multibody system requires computing the generalized accelerations $\dot{\chi}$ for a given vector of generalized forces T and state of the system $\{\vartheta, \chi\}$. The composite-body forward dynamics algorithm described below consists of the following steps: 1) computing the system mass matrix $\mathfrak{M}$, 2) computing the bias vector $\mathcal{C}$, and 3) numerically solving the following linear matrix equation for χ:

$$\mathfrak{M}\dot{\chi} = T - \mathcal{C}   (46)$$

Later in Sec. V we describe the recursive articulated-body forward dynamics algorithm that does not require the explicit computation of either $\mathfrak{M}$ or $\mathcal{C}$.

It is evident from Eq. (46) that the components of the vector $\mathcal{C}$ are the generalized forces for the system when the generalized accelerations $\dot{\chi}$ are all zero. Thus $\mathcal{C}$ can be computed using the inverse dynamics algorithm in Eq. (45). We describe next an efficient composite-body-based recursive algorithm for the computation of the mass matrix $\mathfrak{M}$. This algorithm is based on the following lemma, which contains a decomposition of the mass matrix into block diagonal, block upper triangular, and block lower triangular components.

Lemma 1: Define the composite body inertias $R(k) \in \mathcal{R}^{\overline{\mathfrak{N}}(k) \times \overline{\mathfrak{N}}(k)}$ recursively for all of the bodies in the serial chain as follows:

$$R(0) = 0$$

for $k = 1, \ldots, N$

$$R(k) = \Phi(k, k-1)R(k-1)\Phi^*(k, k-1) + M_m(k)$$

end loop   (47)

Also define

$$R \triangleq \text{diag}\{R(k)\} \in \mathcal{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}}$$

Then we have the following spatial operator decomposition

$$\Phi M_m \Phi^* = R + \tilde{\Phi}R + R\tilde{\Phi}^*   (48)$$

where $\tilde{\Phi} \triangleq \Phi - I$.

Proof: See Appendix A.   □

Physically, $R(k)$ is the modal mass matrix of the composite body formed from all of the bodies outboard of the kth hinge by freezing all of their (deformation plus hinge) degrees of freedom. It follows from Eq. (43) and Lemma 1 that

$$\mathfrak{M} = \mathcal{HC}\Phi M_m \Phi^* \mathcal{HC}^* = \mathcal{HC}R\mathcal{HC}^* + \mathcal{HC}\tilde{\Phi}R\mathcal{HC}^* + \mathcal{HC}R\tilde{\Phi}^*\mathcal{HC}^*$$
$$(49)$$

Note that the three terms on the right of Eq. (49) are block diagonal, block lower triangular, and block upper triangular, respectively. The following algorithm for computing the mass matrix $\mathfrak{M}$ computes the elements of these terms recursively:

$$R(0) = 0$$

for $k = 1, \ldots, N$

$$R(k) = \Phi(k, k-1)R(k-1)\Phi^*(k, k-1) + M_m(k)$$

$$= \mathcal{Q}(k)\phi(t_k, k-1)R^{rr}(k-1)\phi^*(t_k, k-1)\mathcal{Q}^*(k)$$

$$+ M_m(k)$$

$$X(k) = R(k)\mathcal{HC}^*(k)$$

$$\mathfrak{M}(k, k) = \mathcal{HC}(k)X(k)$$

for $j = (k+1), \ldots, N$

$$X(j) = \Phi(j, j-1)X(j-1)$$

$$= \mathcal{Q}(j)\phi(t_j, j-1)X^r(j-1)$$

$$\mathfrak{M}(j, k) = \mathfrak{M}^*(k, j) = \mathcal{HC}(j)X(j)$$

end loop

end loop   (50)

The main recursion proceeds from tip to base and computes the blocks along the diagonal of $\mathfrak{M}$. As each such diagonal element is computed, a new recursion to compute the off-diagonal elements is spawned. The structure of this algorithm closely resembles the composite rigid-body algorithm for computing the mass matrix of rigid multibody systems.[8,12] Like the latter, it is also highly efficient. Additional computational simplifications of the algorithm arising from the sparsity of both $\mathcal{HC}_f(k)$ and $\mathcal{HC}_r(k)$ are easy to incorporate.

V. Factorization and Inversion of the Mass Matrix

An operator factorization of the system mass matrix $\mathfrak{M}$, denoted the "innovations operator factorization," is derived in this section. This factorization is an alternative to the Newton-Euler factorization in Eq. (43) and, in contrast with the latter, the factors in the innovations factorization are square and invertible. Operator expressions for the inverse of these factors are developed and these immediately lead to an operator expression for the inverse of the mass matrix. The operator factorization and inversion results here closely resemble the corresponding results for rigid multibody systems (see Ref. 1).

Given below is a recursive algorithm that defines some required articulated-body quantities:

$$P^+(0) = 0$$

for $k = 1, \ldots, N$

$$P(k) = \Phi(k, k-1)P^+(k-1)\Phi^*(k, k-1)$$
$$+ M_m(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \overline{\mathfrak{N}(k)}}$$

$$D(k) = \mathcal{H}(k)P(k)\mathcal{H}^*(k) \in \mathfrak{R}^{\mathfrak{N}(k) \times \mathfrak{N}(k)}$$

$$G(k) = P(k)\mathcal{H}^*(k)D^{-1}(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \mathfrak{N}(k)}$$

$$K(k+1, k) = \Phi(k+1, k)G(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \mathfrak{N}(k)}$$

$$\bar{\tau}(k) = I - G(k)\mathcal{H}(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \overline{\mathfrak{N}(k)}}$$

$$P^+(k) = \bar{\tau}(k)P(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \overline{\mathfrak{N}(k)}}$$

$$\Psi(k+1, k) = \Phi(k+1, k)\bar{\tau}(k) \in \mathfrak{R}^{\overline{\mathfrak{N}(k)} \times \overline{\mathfrak{N}(k)}}$$

end loop $\qquad\qquad$ (51)

The operator $P \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}}$ is defined as a block diagonal matrix with the kth diagonal element being $P(k)$. The quantities defined in Eq. (51) form the component elements of the following spatial operators:

$$D \triangleq \mathcal{H}P\mathcal{H}^* = \text{diag}\{D(k)\} \in \mathfrak{R}^{\mathfrak{N} \times \mathfrak{N}}$$

$$G \triangleq P\mathcal{H}^*D^{-1} = \text{diag}\{G(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \mathfrak{N}}$$

$$K \triangleq \mathcal{E}_\Phi G \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \mathfrak{N}}$$

$$\bar{\tau} \triangleq I - G\mathcal{H} = \text{diag}\{\bar{\tau}(k)\} \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}}$$

$$\mathcal{E}_\Psi \triangleq \mathcal{E}_\Phi \bar{\tau} \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}} \qquad\qquad (52)$$

The only nonzero block elements of K and $\mathcal{E}_\Psi$ are $K(k+1, k)$ and $\Psi(k+1, k)$, respectively, along the first subdiagonal.

As in the case for $\mathcal{E}_\Phi$, $\mathcal{E}_\Psi$ is nilpotent, so we can define the operator Ψ as follows:

$$\Psi \triangleq (I - \mathcal{E}_\Psi)^{-1} = \begin{bmatrix} I & 0 & \cdots & 0 \\ \Psi(2,1) & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Psi(N,1) & \Psi(N,2) & \cdots & I \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}}$$

$$\qquad\qquad (53)$$

where

$$\Psi(i, j) \triangleq \Psi(i, i-1) \cdots \Psi(j+1, j) \quad \text{for} \quad i > j$$

The structure of the operators $\mathcal{E}_\Psi$ and Ψ is identical to that of the operators $\mathcal{E}_\Phi$ and Φ except that the component elements are now $\Psi(i, j)$ rather than $\Phi(i, j)$. Also, the elements of Ψ have the same semigroup properties as the elements of the operator Φ, and as a consequence, high-level operator expressions involving them can be directly mapped into recursive algorithms,

and the explicit computation of the elements of the operator Ψ is not required.

The innovations operator factorization of the mass matrix is defined in the following lemma.
Lemma 2:

$$\mathfrak{M} = [I + \mathcal{H}\Phi K]D[I + \mathcal{H}\Phi K]^* \qquad\qquad (54)$$

Proof: See Appendix A. $\qquad\qquad \square$

Note that the factor $[I + \mathcal{H}\Phi K] \in \mathfrak{R}^{\overline{\mathfrak{N}} \times \overline{\mathfrak{N}}}$ is square, block lower triangular and nonsingular, whereas D is a block diagonal matrix. This factorization provides a closed-form expression for the block LDL^* decomposition of $\mathfrak{M}$. The following lemma gives the closed-form operator expression for the inverse of the factor $[I + \mathcal{H}\Phi K]$.
Lemma 3:

$$[I + \mathcal{H}\Phi K]^{-1} = [I - \mathcal{H}\Psi K] \qquad\qquad (55)$$

Proof: See Appendix A. $\qquad\qquad \square$

It follows from Lemmas 2 and 3 that the operator expression for the inverse of the mass matrix is given by Lemma 4.
Lemma 4:

$$\mathfrak{M}^{-1} = [I - \mathcal{H}\Psi K]^*D^{-1}[I - \mathcal{H}\Psi K] \qquad\qquad (56)$$

Once again, note that the factor $[I - \mathcal{H}\Psi K]$ is square, block lower triangular, and nonsingular, and so Lemma 4 provides a closed-form expression for the block LDL^* decomposition of $\mathfrak{M}^{-1}$.

VI. Articulated-Body Forward Dynamics Algorithm

We first use the operator expression for the mass matrix inverse developed in Sec. V to obtain an operator expression for the generalized accelerations $\dot{\chi}$. This expression directly leads to a recursive algorithm for the forward dynamics of the system. The structure of this algorithm is completely identical in form to the articulated-body algorithm for serial rigid multibody systems. The computational cost of this algorithm is further reduced by separately processing the flexible and hinge degrees of freedom at each step in the recursion, and this leads to the articulated-body forward dynamics algorithm for serial flexible multibody systems. This algorithm is an alternative to the composite-body forward dynamics algorithm developed earlier.

The following lemma describes the operator expression for the generalized accelerations $\dot{\chi}$ in terms of the generalized forces T.
Lemma 5:

$$\dot{\chi} = [I - \mathcal{H}\Psi K]^*D^{-1}\big[T - \mathcal{H}\Psi\{KT + Pa_m + b_m + K_m\vartheta\}\big]$$
$$- K^*\Psi^*a_m \qquad\qquad (57)$$

Proof: See Appendix A. $\qquad\qquad \square$

As in the case of rigid multibody systems,[1,2] the direct recursive implementation of Eq. (57) leads to the following recursive forward dynamics algorithm:

$$z^+(0) = 0$$

for $k = 1, \ldots, N$

$$z(k) = \Phi(k, k-1)z^+(k-1) + P(k)a_m(k) + b_m(k)$$
$$+ K_m(k)\vartheta(k)$$

$$\epsilon(k) = T(k) - \mathcal{H}(k)z(k)$$

$$\nu(k) = D^{-1}(k)\epsilon(k)$$

$$z^+(k) = z(k) + G(k)\epsilon(k)$$

end loop $\qquad\qquad$ (58)

$$\alpha_m(N+1) = 0$$

for $k = N, \dots, 1$

$$\alpha_m^+(k) = \Phi^*(k+1,k)\alpha_m(k+1)$$

$$\dot\chi(k) = \nu(k) - G^*(k)\alpha_m^+(k)$$

$$\alpha_m(k) = \alpha_m^+(k) + \mathcal{K}^*(k)\dot\chi(k) + a_m(k)$$

end loop

The structure of this algorithm is closely related to the structure of the well-known Kalman filtering and smoothing algorithms.[8] All of the degrees of freedom for each body [as characterized by the joint map matrix $\mathcal{K}^*(.)$] are processed together at each recursion step in this algorithm. However, by taking advantage of the sparsity and special structure of the joint map matrix, additional reduction in computational cost is obtained by processing the flexible degrees of freedom and the hinge degrees of freedom separately. These simplifications are described in the following sections.

A. Simplified Algorithm for the Articulated-Body Quantities

Instead of a detailed derivation, we describe here the conceptual basis for the separation of the modal and hinge degrees of freedom for each body. First we recall the velocity recursion equation in Eq. (11)

$$V_m(k) = \Phi^*(k+1,k)V_m(k+1) + \mathcal{K}^*(k)\chi(k) \tag{59}$$

and the partitioned form of $\mathcal{K}(k)$ in Eq. (13)

$$\mathcal{K}(k) = \begin{bmatrix} \mathcal{K}_f(k) \\ \mathcal{K}_r(k) \end{bmatrix} \tag{60}$$

Introducing a dummy variable k', we can rewrite Eq. (59) as

$$V_m(k') = \Phi^*(k+1,k')V_m(k+1) + \mathcal{K}_f^*(k)\dot\eta(k)$$

$$V_m(k) = \Phi^*(k',k)V_m(k') + \mathcal{K}_r^*(k)\beta(k) \tag{61}$$

where

$$\Phi(k+1,k') \triangleq \Phi(k+1,k) \quad \text{and} \quad \Phi(k',k) \triangleq I$$

Conceptually, each flexible body is now associated with two new bodies. The first one has the same kinematical and mass/inertia properties as the real body and has the flexible degrees

of freedom. The second body is a fictitious body that is massless and has zero extent. It is associated with the hinge degrees of freedom. The serial chain now contains twice the number of bodies as the original one with half of the new bodies being fictitious ones. The new $\mathcal{K}^*$ operator now has the same number of columns but twice the number of rows as the original $\mathcal{K}^*$ operator. The new Φ operator has twice as many rows and columns as the original one. Repeating the analysis described in the preceding sections, we once again obtain the same operator expression as Eq. (57). This expression also leads to a recursive forward dynamics algorithm as in Eq. (58). However, each sweep in the algorithm now contains twice as many steps as the original algorithm. But since each step now processes only a smaller number of degrees of freedom, this leads to a reduction in the overall cost. The new algorithm [replacing Eq. (51)] for computing the articulated body quantities is as follows:

$$P^+(0) = 0$$

for $k = 1, \dots, N$

$$\Gamma(k) = \mathfrak{B}(k,k-1)P^+(k-1)\mathfrak{B}^*(k,k-1) \in \mathfrak{R}^{6\times6}$$

$$P(k) = \mathfrak{A}(k)\Gamma(k)\mathfrak{A}^*(k) + M_m(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

$$D_f(k) = \mathcal{K}_f(k)P(k)\mathcal{K}_f^*(k) \in \mathfrak{R}^{nm(k)\times nm(k)}$$

$$G_f(k) = P(k)\mathcal{K}_f^*(k)D_f^{-1}(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times nm(k)}$$

$$\bar\tau_f(k) = I - G_f(k)\mathcal{K}_f(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

$$P_r(k) = \bar\tau_f(k)P(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

$$D_r(k) = \mathcal{K}_r(k)P_r(k)\mathcal{K}_r^*(k) \in \mathfrak{R}^{nr(k)\times nr(k)}$$

$$G_r(k) = P_r(k)\mathcal{K}_r^*(k)D_r^{-1}(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times nr(k)}$$

$$\bar\tau_r(k) = I - G_r(k)\mathcal{K}_r(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

$$P^+(k) = \bar\tau_r(k)P_r(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

$$\Psi(k+1,k) = \Phi(k+1,k)\bar\tau(k) \in \mathfrak{R}^{\overline{\mathfrak{R}}(k)\times\overline{\mathfrak{R}}(k)}$$

end loop $\tag{62}$

We now use the sparsity of $\mathfrak{B}(k+1,k)$, $\mathcal{K}_f(k)$, and $\mathcal{K}_r(k)$ to further simplify the preceding algorithm. Using the symbol "x" to indicate "don't care" blocks, the structure in block partitioned form of some of the quantities in Eq. (62) is given below:

$$\Gamma(k) = \phi(t_k, k-1)P_R^+(k-1)\phi^*(t_k, k-1), \qquad [P_R^+(k) \text{ is defined below}]$$

$$G_f(k) = \begin{pmatrix} \mathsf{x} \\ g(k) \end{pmatrix} \quad \text{where} \quad g(k) = \mu(k)D_f^{-1}(k) \in \mathfrak{R}^{6\times nm(k)}, \qquad \mu(k) \triangleq \begin{bmatrix} P^{rf}(k), & P^{rr}(k) \end{bmatrix}\mathcal{K}_f^*(k) \in \mathfrak{R}^{6\times nm(k)}$$

$$P_r(k) = \begin{pmatrix} \mathsf{x} & \mathsf{x} \\ \mathsf{x} & P_R(k) \end{pmatrix} \quad \text{where} \quad P_R(k) = P^{rr}(k) - g(k)\mu^*(k) \in \mathfrak{R}^{6\times6}$$

$$D_r(k) = H_{\mathfrak{F}}(k)P_R(k)H_{\mathfrak{F}}^*(k) \in \mathfrak{R}^{nr(k)\times nr(k)}$$

$$G_r(k) = \begin{pmatrix} \mathsf{x} \\ G_R(k) \end{pmatrix} \quad \text{where} \quad G_R(k) \triangleq P_R(k)H_{\mathfrak{F}}^*(k)D_r^{-1}(k) \in \mathfrak{R}^{6\times nr(k)}$$

$$\bar\tau_r(k) = \begin{pmatrix} I & \mathsf{x} \\ 0 & \bar\tau_R(k) \end{pmatrix} \quad \text{where} \quad \bar\tau_R(k) = I - G_R(k)H_{\mathfrak{F}}(k) \in \mathfrak{R}^{6\times6}$$

$$P^+(k) = \begin{pmatrix} \mathsf{x} & \mathsf{x} \\ \mathsf{x} & P_R^+(k) \end{pmatrix} \quad \text{where} \quad P_R^+(k) = \bar\tau_R(k)P_R(k) \in \mathfrak{R}^{6\times6}$$

Using the structure just described, the simplified algorithm for computing the articulated body quantities is as follows:

$$P_R^+(0) = 0$$

for $k = 1, \ldots, N$

$$\Gamma(k) = \phi(t_k, k-1)P_R^+(k-1)\phi^*(t_k, k-1)$$

$$P(k) = \mathcal{Q}(k)\Gamma(k)\mathcal{Q}^*(k) + M_m(k)$$

$$D_f(k) = \mathcal{K}_f(k)P(k)\mathcal{K}_f^*(k)$$

$$\mu(k) = \left[P^{rf}(k), \quad P^{rr}(k) \right]\mathcal{K}_f^*(k)$$

$$g(k) = \mu(k)D_f^{-1}(k)$$

$$P_R(k) = P^{rr}(k) - g(k)\mu^*(k)$$

$$D_R(k) = H_{\mathfrak{F}}(k)P_R(k)H_{\mathfrak{F}}^*(k)$$

$$G_R(k) = P_R(k)H_{\mathfrak{F}}^*(k)D_R^{-1}(k)$$

$$\bar{\tau}_R(k) = I - G_R(k)H_{\mathfrak{F}}(k)$$

$$P_R^+(k) = \bar{\tau}_R(k)P_R(k)$$

end loop   (63)

B. Simplified Articulated-Body Forward Dynamics Algorithm

The complete recursive articulated-body forward dynamics algorithm for a serial flexible multibody system follows directly from the recursive implementation of the expression in Eq. (57). The algorithm consists of the following steps: 1) a base-to-tip recursion as in Eq. (45) for computing the modal spatial velocities $V_m(k)$ and the Coriolis and gyroscopic terms $a_m(k)$ and $b_m(k)$ for all of the bodies; 2) computation of the articulated body quantities using Eqs. (B4) and (63); and 3) a tip-to-base recursion followed by a base-to-tip recursion for the joint accelerations $\dot{\chi}$ as described below:

$$z_R^+(0) = 0$$

for $k = 1, \ldots, N$

$$z(k) = \begin{pmatrix} z_f(k) \\ z_r(k) \end{pmatrix}$$

$$= \mathcal{Q}(k)\phi(t_k, k-1)z_R^+(k-1) + b_m(k)$$

$$+ K_m(k)\vartheta(k) \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)}$$

$$\epsilon_f(k) = T_f(k) - z_f(k) + \left[\Pi_{\mathfrak{F}}^d(k)\right]^* z_r(k) \in \mathfrak{R}^{nm(k)}$$

$$\nu_f(k) = D_f^{-1}(k)\epsilon_f(k) \in \mathfrak{R}^{nm(k)}$$

$$z_R(k) = z_r(k) + g(k)\epsilon_f(k) + P_R(k)a_{mR}(k) \in \mathfrak{R}^6$$

$$\epsilon_R(k) = T_R(k) - H_{\mathfrak{F}}(k)z_R(k) \in \mathfrak{R}^{nr(k)}$$

$$\nu_R(k) = D_R^{-1}(k)\epsilon_R(k) \in \mathfrak{R}^{nr(k)}$$

$$z_R^+(k) = z_R(k) + G_R(k)\epsilon_R(k) \in \mathfrak{R}^6$$

end loop   (64)

$$\alpha_m(N+1) = 0$$

for $k = N, \ldots, 1$

$$\alpha_R^+(k) = \phi^*(t_{k+1}, k)\mathcal{Q}^*(k+1)\alpha_m(k+1) \in \mathfrak{R}^6$$

$$\dot\beta(k) = \nu_R(k) - G_R^*(k)\alpha_R^+(k) \in \mathfrak{R}^{nr(k)}$$

$$\alpha_R(k) = \alpha_R^+(k) + H_{\mathfrak{F}}^*(k)\dot\beta(k) + a_{mR}(k) \in \mathfrak{R}^6$$

$$\ddot\eta(k) = \nu_f(k) - g^*(k)\alpha_R(k) \in \mathfrak{R}^{nm(k)}$$

$$\alpha_m(k) = \begin{bmatrix} \ddot\eta(k) \\ \alpha_R(k) - \Pi_{\mathfrak{F}}^d(k)\ddot\eta(k) \end{bmatrix} \in \mathfrak{R}^{\overline{\mathfrak{N}}(k)}$$

end loop

The recursion in Eq. (64) is obtained by simplifying the recursions in Eq. (58) in the same manner as described in Sec. VI.A.

In contrast with the composite-body forward dynamics algorithm described in Sec. IV, the articulated-body forward dynamics algorithm does not require the explicit computation of either $\mathfrak{M}$ or $\mathcal{C}$. The structure of this articulated-body algorithm closely resembles the recursive articulated-body forward dynamics algorithm for rigid multibody systems described in the literature.[1,15]

The articulated-body forward dynamics algorithm has been used to develop a dynamics simulation software package (called DARTS) for the high-speed, real-time, hardware-in-the-loop simulation of planetary spacecraft. Validation of the DARTS software was carried out by comparing simulation results with those from a standard, flexible, multibody simulation package.[6] The results from the two independent simulations have shown complete agreement.

VII. Computational Cost

This section discusses the computational cost of the composite-body and the articulated-body forward dynamics algorithms. For low-spin multibody systems, it has been suggested in Ref. 16 that using ruthlessly linearized models for each flexible body can lead to significant computational reduction without sacrificing fidelity. These linearized models are considerably less complex and do not require much of the modal integral data for the individual flexible bodies. All computational costs given below are based on the use of ruthlessly linearized models and the computationally simplified steps described in Appendix B.

Flexible multibody systems typically involve both rigid and flexible bodies and, in addition, different sets of modes are used to model the flexibility of each body. As a consequence, where possible, we describe the contribution of a typical (nonextremal) flexible body, denoted the kth body, to the overall computational cost. Note that the computational cost for extremal bodies as well as for rigid bodies is lower than that for a nonextremal flexible body. Summing up this cost for all the bodies in the system gives a figure close to the true computational cost for the algorithm. Without any loss in generality, we have assumed here that all of the hinges are single-degree-of-freedom rotary joints and that free-free assumed modes are being used. The computational costs are given in the form of polynomial expressions for the number of floating point operations, with the symbol M denoting multiplications and A denoting additions.

A. Computational Cost of the Composite-Body Forward Dynamics Algorithm

The composite-body forward dynamics algorithm described in Sec. IV is based on solving the linear matrix equation

$$\mathfrak{M}\dot{\chi} = T - \mathcal{C}$$

The computational cost of this forward dynamics algorithm is given below:

1) Cost of computing $R(k)$ for the kth body using the algorithm in Eq. (50) is

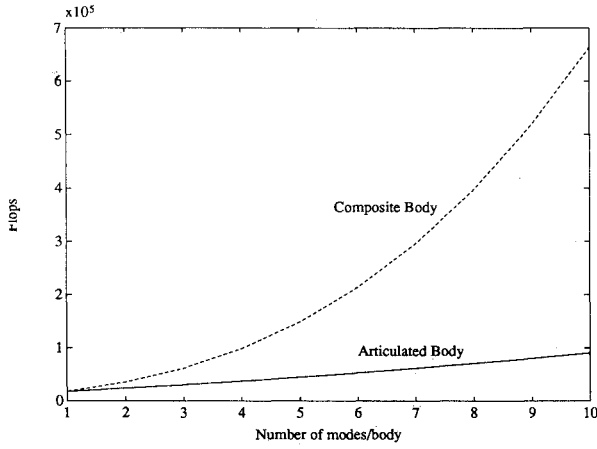$$\left[48n_m(k) + 90\right]M + \left[N_m^2(k) + \frac{97}{2}n_m(k) + 116\right]A$$

Fig. 2 Comparison of the computational cost in floating point operations for the articulated- and composite-body forward dynamics algorithms for serial-chain multibody systems with 10 flexible bodies.

2) Contribution of the kth body to the cost of computing $\mathfrak{M}$ [excluding cost of $R(k)$] using the algorithm in Eq. (50) is

$$\left\{k\left[12n_m^2(k)+34n_m(k)+13\right]\right\}M$$

$$+\left\{k\left[11n_m^2(k)+24n_m(k)+13\right]\right\}A$$

3) Setting the generalized accelerations $\dot{\chi}=0$, the vector $\mathcal{C}$ can be obtained by using the inverse dynamics algorithm described in Eq. (45) for computing the generalized forces T. The contribution of the kth body to the computational cost for $\mathcal{C}(k)$ is

$$\left\{2n_m^2(k)+54n_m(k)+206\right\}M+\left\{2n_m^2(k)+50n_m(k)+143\right\}A$$

4) The cost of computing $T-\mathcal{C}$ is $\{\mathfrak{N}\}A$.

5) The cost of solving the linear equation in Eq. (46) for the accelerations $\dot{\chi}$ is

$$\left\{\frac{1}{6}\mathfrak{N}^3+\frac{3}{2}\mathfrak{N}^2-\frac{2}{3}\mathfrak{N}\right\}M+\left\{\frac{1}{6}\mathfrak{N}^3+\mathfrak{N}^2-\frac{7}{6}\mathfrak{N}\right\}A$$

The overall complexity of the composite-body forward dynamics algorithm is $O(\mathfrak{N}^3)$.

B. Computational Cost of the Articulated-Body Forward Dynamics Algorithm

The articulated-body forward dynamics algorithm is based on the recursions described in Eqs. (B4), (63), and (64). Since the computations in Eq. (B4) can be carried out prior to the dynamics simulation, the cost of this recursion is not included in the cost of the overall forward dynamics algorithm described below:

1) The algorithm for the computation of the articulated body quantities is given in Eq. (63). The step involving the computation of $D_f^{-1}(k)$ can be carried out either by an explicit inversion of $D_f(k)$ with $O[n_m^3(k)]$ cost, or by the indirect procedure described in Eq. (B3) with $O[n_m^2(k)]$ cost. The first method is more efficient than the second one for $n_m(k)\le 7$.

a) Cost of Eq. (63) for the kth body based on the explicit inversion of $D_f(k)$ [used when $n_m(k)\le 7$] is

$$\left\{(5/6)n_m^3(k)+(25/2)n_m^2(k)+(764/3)n_m(k)+180\right\}M$$

$$+\left\{(5/6)n_m^3(k)+(21/2)n_m^2(k)+(548/3)n_m(k)+164\right\}A$$

b) Cost of Eq. (63) for the kth body based on the indirect computation of $D_f^{-1}(k)$ [used when $n_m(k)\ge 8$] is

$$\left\{12n_m^2(k)+255n_m(k)+572\right\}M$$

$$+\left\{13n_m^2(k)+182n_m(k)+445\right\}A$$

2) The cost for the tip-to-base recursion sweep in Eq. (64) for the kth body is

$$\left\{n_m^2(k)+25n_m(k)+49\right\}M+\left\{n_m^2(k)+24n_m(k)+50\right\}A$$

3) The cost for the base-to-tip recursion sweep in Eq. (64) for the kth body is

$$\left\{18n_m(k)+52\right\}M+\left\{19n_m(k)+42\right\}A$$

The overall complexity of this algorithm is $O(Nn_m^2)$, where n_m is an upper bound on the number of modes per body in the system.

From a comparison of the computational costs, it is clear that the articulated-body algorithm is more efficient than the composite-body algorithm as the number modes and bodies in the multibody system increases. Figure 2 contains a plot of the computational cost (in floating point operations) of the composite-body and the articulated-body forward dynamics algorithms vs the number of assumed-modes per body for a serial chain with 10 flexible bodies. The articulated-body algorithm is faster by over a factor of 3 for 5 modes per body, and by over a factor of 7 for the case of 10 modes per body. The divergence between the costs for the two algorithms becomes even more rapid as the number of bodies is increased.

VIII. Extensions to General Topology Flexible Multibody Systems

For rigid multibody systems, Ref. 11 describes the extensions to the dynamics formulation and the algorithms that are required as the toplogy of the system goes from a serial chain topology, to a tree topology, and finally to a closed-chain topology system. The key to this progression is the invariance of the operator description of the system dynamics to increases in the topological complexity of the system. Indeed, as seen here, the operator description of the dynamics remains the same even when the multibody system contains flexible rather than rigid component bodies. Thus, using the approach in Ref. 11 for rigid multibody systems, the dynamics formulation and algorithms for flexible multibody systems with serial topology can be extended in a straightforward manner to systems with tree or closed-chain topology. Based on these observations, extending the serial chain dynamics algorithms described in this paper to tree topology flexible multibody systems requires the following steps:

1) For each outward sweep involving a base-to-tip(s) recursion, at each body, the outward recursion must be continued along each outgoing branch emanating from the current body.

2) For each inward sweep involving a tip(s)-to-base recursion, at each body, the recursion must be continued inward only after summing up contributions from each of the other incoming branches for the body.

A closed-chain topology flexible multibody system can be regarded as a tree topology system with additional closure constraints. As described in Ref. 11, the dynamics algorithm for closed-chain systems consists of recursions involving the dynamics of the tree topology system, and in addition the computation of the closure constraint forces. The computation of the constraint forces requires the effective inertia of the tree topology system reflected to the points of closure. The algorithm for closed-chain flexible multibody systems for computing these inertias is identical in form to the recursive algorithm described in Ref. 11.

IX. Conclusions

This paper uses spatial operator methods to develop a new dynamics formulation for flexible multibody systems. A key feature of the formulation is that the operator description of the flexible system dynamics is identical in form to the corresponding operator description of the dynamics of rigid multibody systems. A significant advantage of this unifying

approach is that it allows ideas and techniques for rigid multibody systems to be easily applied to flexible multibody systems. The Newton-Euler operator factorization of the mass matrix forms the basis for recursive algorithms such as those for the inverse dynamics, the computation of the mass matrix, and the composite-body forward dynamics algorithm for the flexible multibody system. Subsequently, we develop the articulated-body forward dynamics algorithm, which, in contrast to the composite-body forward dynamics algorithm, does not require the explicit computation of the mass matrix. While the computational cost of the algorithms depends on factors such as the topology and the amount of flexibility in the multibody system, in general, the articulated-body forward dynamics algorithm is by far the more efficient algorithm for flexible multibody systems containing even a small number of flexible bodies. All of the algorithms are closely related to those encountered in the domain of Kalman filtering and smoothing. Whereas the major focus in this paper is on flexible multibody systems with serial chain topology, the extensions to tree and closed-chain topologies are straightforward and are described as well.

Appendix A: Proofs of the Lemmas

At the operator level, the proofs of the lemmas in this paper are completely analogous to those for rigid multibody systems.[1,2]

Proof of Lemma 1

Using operators, we can rewrite Eq. (47) in the form

$$M_m = R - \mathcal{E}_\Phi R \mathcal{E}_\Phi^* \tag{A1}$$

From Eq. (19) it follows that $\Phi \mathcal{E}_\Phi = \mathcal{E}_\Phi \Phi = \Phi - I = \tilde{\Phi}$. Multiplying Eq. (A1) from the left and right by Φ and Φ^* respectively leads to

$$\Phi M_m \Phi^* = \Phi R \Phi^* - \Phi \mathcal{E}_\Phi R \mathcal{E}_\Phi^* \Phi^*$$

$$= (\tilde{\Phi} + I) R (\tilde{\Phi} + I)^* - \tilde{\Phi} R \tilde{\Phi}^* = R + \tilde{\Phi} R + R \tilde{\Phi}^*$$

Proof of Lemma 2

It is easy to verify that $\tilde{\tau} P \tilde{\tau}^* = \tilde{\tau} P$. As a consequence, the recursion for $P(.)$ in Eq. (51) can be rewritten in the form

$$M_m = P - \mathcal{E}_\Psi P \mathcal{E}_\Psi^* = P - \mathcal{E}_\Psi P \mathcal{E}_\Phi^* = P - \mathcal{E}_\Phi P \mathcal{E}_\Phi^* + KDK^* \tag{A2}$$

Pre- and post-multiplying Eq. (A2) by Φ and Φ^* then leads to

$$\Phi M_m \Phi^* = P + \tilde{\Phi} P + P \tilde{\Phi}^* + \Phi KDK^* \Phi^*$$

Hence,

$$\Rightarrow \mathfrak{M} = \mathcal{K} \Phi M_m \Phi^* \mathcal{K}^* = \mathcal{K}[P + \tilde{\Phi} P + P \tilde{\Phi}^* + \Phi KDK^* \Phi^*] \mathcal{K}^*$$

$$= D + \mathcal{K} \Phi KD + DK^* \Phi^* \mathcal{K}^* + \mathcal{K} \Phi KDK^* \Phi^* \mathcal{K}^*$$

$$= [I + \mathcal{K} \Phi K] D [I + \mathcal{K} \Phi K]^*$$

Proof of Lemma 3

Using a standard matrix identity, we have that

$$[I + \mathcal{K} \Phi K]^{-1} = I - \mathcal{K} \Phi [I + K \mathcal{K} \Phi]^{-1} K \tag{A3}$$

Note that

$$\Psi^{-1} = I - \mathcal{E}_\Psi = (I - \mathcal{E}_\Phi) + \mathcal{E}_\Phi G \mathcal{K} = \Phi^{-1} + K \mathcal{K} \tag{A4}$$

from which it follows that

$$\Psi^{-1} \Phi = I + K \mathcal{K} \Phi$$

Using this with Eq. (A3), it follows that

$$[I + \mathcal{K} \Phi K]^{-1} = I - \mathcal{K} \Phi [\Psi^{-1} \Phi]^{-1} K = I - \mathcal{K} \Psi K$$

Proof of Lemma 5

From Eqs. (42) and (43), the expression for the generalized accelerations $\dot{\chi}$ is given by

$$\dot{\chi} = \mathfrak{M}^{-1}(T - \mathcal{C})$$

$$= [I - \mathcal{K} \Psi K]^* D^{-1} [I - \mathcal{K} \Psi K]$$

$$\times \left[T - \mathcal{K} \Phi [M_m \Phi^* a_m + b_m + K_m \vartheta] \right] \tag{A5}$$

From Eq. (A4) we have that

$$[I - \mathcal{K} \Psi K] \mathcal{K} \Phi = \mathcal{K} \Psi [\Psi^{-1} - k \mathcal{K}] \Phi = \mathcal{K} \Psi \tag{A6}$$

Thus Eq. (A5) can be written as

$$\dot{\chi} = [I - \mathcal{K} \Psi K]^* D^{-1} \left[T - \mathcal{K} \Psi [KT + M_m \Phi^* a_m + b_m + K_m \vartheta] \right] \tag{A7}$$

From Eq. (A2) it follows that

$$M_m = P - \mathcal{E}_\Psi P \mathcal{E}_\Phi^* \quad \Rightarrow \quad \Psi M_m \Phi^* = \Psi P + P \tilde{\Phi}^* \tag{A8}$$

and so Eq. (A7) simplifies to

$$\dot{\chi} = [I - \mathcal{K} \Psi K]^* D^{-1} \left[T - \mathcal{K} \Psi [KT + P a_m + b_m + K_m \vartheta] \right.$$

$$\left. - \mathcal{K} P \tilde{\Phi}^* a_m \right] \tag{A9}$$

From Eq. (A4) we have that

$$[I - \mathcal{K} \Psi K]^* D^{-1} \mathcal{K} P \tilde{\Phi}^* = [I - \mathcal{K} \Psi K]^* K^* \Phi^*$$

$$= K^* \Psi^* [\Psi^{-*} - K \mathcal{K}]^* \Phi^* = K^* \Psi^* \tag{A10}$$

Using this in Eq. (A9) leads to the result.

Appendix B: Ruthless Linearization of Flexible Body Dynamics

It has been pointed out in recent literature[16,17] that the use of modes for modeling body flexibility leads to "premature linearization" of the dynamics in the sense that, while the dynamics model contains deformation-dependent terms, the geometric stiffening terms are missing. These missing geometric stiffening terms are the dominant terms among the first-order (deformation) dependent terms. In general, it is necessary to take additional steps to recover the missing geometric stiffness terms to obtain a "consistently" linearized model with the proper degree of fidelity.

However for systems with low spin rate, there is typically little loss in model fidelity when the deformation- and deformation rate-dependent terms are dropped altogether from the dynamical equations of motion.[16] Such models have been dubbed the "ruthlessly linearized models." These linearized models are considerably less complex and do not require most of the modal integrals data for each individual flexible body. In this model, the approximations to $M_m(k)$, $a_m(k)$, and $b_m(k)$ are as follows:

$$M_m(k) \approx M_m^0(k), \quad a_m(k) \approx \begin{bmatrix} 0 \\ a_{mR}^0(k) \end{bmatrix}, \quad b_m(k) \approx b_m^0(k) \tag{B1}$$

With this approximation, $M_m(k)$ is constant in the body frame, whereas $a_m(k)$ and $b_m(k)$ are independent of $\eta(k)$

and $\dot{\eta}(k)$. With this being the case, the formation of D_f^{-1} in Eq. (63) can be simplified. Using the matrix identity

$$[A + BCB^*]^{-1} = A^{-1} - A^{-1}B[C^{-1} + B^*A^{-1}B]^{-1}B^*A^{-1} \tag{B2}$$

which holds for general matrices A, B, and C, it is easy to verify that

$$D_f^{-1}(k) = \Lambda(k) - \Upsilon(k)[\Gamma^{-1}(k) + \Omega(k)]^{-1}(k)\Upsilon^*(k) \tag{B3}$$

where the matrices $\Lambda(k)$, $\Omega(k)$, and $\Upsilon(k)$ are precomputed just once prior to the dynamical simulation as follows:

for $k = 1, \ldots, N$

$$\Lambda(k) = \left[\mathcal{H}_f(k)M_m(k)\mathcal{H}_f^*(k)\right]^{-1} \in \mathcal{R}^{\mathfrak{N} \times \mathfrak{N}}$$

$$\zeta(k) = \mathcal{H}_f(k)\mathcal{Q}(k) \in \mathcal{R}^{\mathfrak{N} \times 6}$$

$$\Upsilon(k) = \Lambda(k)\zeta(k) \in \mathcal{R}^{\mathfrak{N} \times 6}$$

$$\Omega(k) = \zeta^*(k)\Upsilon(k) \in \mathcal{R}^{6 \times 6}$$

end loop   (B4)

The use of Eq. (B3) reduces the computational cost for computing the articulated-body inertias to a quadratic rather than a cubic function of the number of modes.

Acknowledgments

References

[1]Rodriguez, G., Kreutz, K., and Jain, A., "A Spatial Operator Algebra for Manipulator Modeling and Control," *International Journal of Robotics Research,* Vol. 10, No. 4. 1991, pp. 371–381.

[2]Jain, A., "Unified Formulation of Dynamics for Serial Rigid Multibody Systems," *Journal of Guidance, Control, and Dynamics,* Vol. 14, No. 3, 1991, pp. 531–542.

[3]Kim, S. S., and Haug, E. J., "A Recursive Formulation for Flexible Multibody Dynamics, Part I: Open-Loop Systems," *Computer Methods in Applied Mechanics and Engineering,* Vol. 71, No. 3, 1988, pp. 293–314.

[4]Changizi, K., and Shabana, A. A., "A Recursive Formulation for the Dynamic Analysis of Open Loop Deformable Multibody Systems," *Journal of Applied Mechanics,* Vol. 55, No. 3, 1988, pp. 687–693.

[5]Keat, J. E., "Multibody System Order n Dynamics Formulation Based on Velocity Transform Method," *Journal of Guidance, Control, and Dynamics,* Vol. 13, No. 2, 1990, pp. 207–212.

[6]Bodley, C. S., Devers, A. D., Park, A. C., and Frisch, H. P., "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structure (DISCOS)," NASA TP 1219, May 1978.

[7]Singh, R. P., VanderVoort, R. J., and Likins, P. W., "Dynamics of Flexible Bodies in Tree Topology—A Computer-Oriented Approach," *Journal of Guidance, Control, and Dynamics,* Vol. 8, No. 5, 1985, pp. 584–590.

[8]Rodriguez, G., "Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics," *Journal of Robotics and Automation,* Vol. 3, No. 6, 1987, pp. 624–639.

[9]Likins, P. W., "Modal Method for Analysis of Free Rotations of Spacecraft," *AIAA Journal,* Vol. 5, No. 7, 1967, pp. 1304–1308.

[10]Jain, A., and Rodriguez, G., "Recursive Dynamics for Flexible Multibody Systems Using Spatial Operators," Jet Propulsion Lab., JPL Pub. 90-26, Pasadena, CA, Dec. 1990.

[11]Rodriguez, G., Jain, A., and Kreutz, K., "Spatial Operator Algebra for Multibody System Dynamics," *Journal of the Astronautical Sciences,* Vol. 40, No. 1, 1992, pp. 27–50.

[12]Walker, M. W., and Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *Journal of Dynamic Systems, Measurement, and Control,* Vol. 104, No. 3, 1982, pp. 205–211.

[13]Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., "On-line Computational Scheme for Mechanical Manipulators," *Journal of Dynamic Systems, Measurement, and Control,* Vol. 102, No. 2, 1980, pp. 69–76.

[14]Jain, A., and Rodriguez, G., "Kinematics and Dynamics of Under-Actuated Manipulators," IEEE International Conference on Robotics and Automation, Sacramento, CA, April 1991, pp. 1754–1759.

[15]Featherstone, R., "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *International Journal of Robotics Research,* Vol. 2, No. 1, 1983, pp. 13–30.

[16]Padilla, C. E., and von Flotow, A. H., "Nonlinear Strain-Displacement Relations and Flexible Multibody Dynamics," *Proceedings of the 3rd Annual Conference on Aerospace Computational Control* (Oxnard, CA), Vol. 1, Jet Propulsion Lab., JPL Pub. 89-45, Pasadena, CA, 1989, pp. 230–245.

[17]Kane, T. R., Ryan, R. R., and Banerjee, A. K., "Dynamics of a Cantilevered Beam Attached to a Moving Base," *Journal of Guidance, Control, and Dynamics,* Vol. 10, No. 2, 1987, pp. 139–151.