N94-14631

Spatial Operator Algebra for Flexible Multibody Dynamics

A. Jain and G. Rodriguez
Jet Propulsion Laboratory/California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

Abstract

This paper presents an approach to modeling the dynamics of flexible multibody systems such as flexible spacecraft and limber space robotic systems. A large number of degrees of freedom and complex dynamic interactions are typical in these systems. This paper uses spatial operators to develop efficient recursive algorithms for the dynamics of these systems. This approach very efficiently manages complexity by means of a hierarchy of mathematical operations.

1. Introduction

A wide range of complex mechanical systems can be modeled as a set of hinge-connected flexible and rigid bodies. This paper presents an approach to modeling the dynamics of such systems that uses spatial operators. This approach very efficiently manages complexity by means of a hierarchy of mathematical operations. The highest level in this hierarchy consists of spatial operators which relate velocities, accelerations, and forces between distinct points in the system. At lower levels, each spatial operator is decomposed easily into detailed spatially recursive algorithms to do computation. The recursive algorithms are cast within the highly developed framework of filtering and smoothing theory. Algorithms which are quite popular in state estimation theory for discrete-time systems can now be applied to conduct spatially recursive operations essential in multibody dynamics. The main focus is on serial chains, but extensions to general topologies are also described. Comparison of computational costs illustrates the efficiency of the recursive algorithms.

This paper uses spatial operators [1,2] to develop efficient recursive algorithms for flexible multibody systems for such applications as flexible spacecraft and limber space robotic systems. A large number of degrees of freedom and complex dynamic interactions are typical in these systems. The main contributions of the paper are: (1) high-level architectural understanding of the mass matrix and its inverse, (2) high-level expressions which can be easily implemented with spatial Kalman filtering and smoothing, (3) efficient inverse and forward dynamics recursive algorithms, and (4) analysis of computational cost of the new algorithms. This adds to the rapidly developing body of research in the recursive dynamics of flexible multibody systems [3–5].

2. Equations of Motion

Equations of motion are developed for a serial system formed by N articulated flexible bodies. Recursive relationships between the modal velocities, accelerations and forces are developed. Spatial operators express these relationships compactly to obtain what is referred to here as the Newton-Euler mass matrix factorization.

Each flexible body has a *lumped mass* model formed by a set of nodal rigid bodies. Such models are typically developed using standard finite element analysis. The k^{th} body has $n_s(k)$ nodes. The j^{th} node on the k^{th} body is called the j_k^{th} node. There is a *body reference frame* $\mathcal{F}_k$ for the k^{th} body. Deformation of the nodes on the body is described with respect to this body reference frame, while the rigid–body motion of the k^{th} body is characterized by the motion of frame $\mathcal{F}_k$.

The 6-dimensional *spatial deformation* (slope plus translational) of node j_k (with respect to frame $\mathcal{F}_k$) is $u(j_k) \in \Re^6$. The overall deformation field for the k^{th} body is the vector $u(k) = \text{col}\left\{u(j_k)\right\} \in \Re^{6n_s(k)}$. The vector from frame $\mathcal{F}_k$ to the reference frame on node j_k is $l(k, j_k) \in \Re^3$.

The spatial inertia of the j^{th} node is

$$M_s(j_k) = \begin{pmatrix} \mathcal{J}(j_k) & m(j_k)\tilde{p}(j_k) \\ -m(j_k)\tilde{p}(j_k) & m(j_k)I \end{pmatrix} \in \Re^{6\times 6} \tag{1}$$

where $\mathcal{J}(j_k)$, $p(j_k)$ and $m(j_k)$ are the inertia tensor about the node reference frame, the vector from the node reference frame to its center of mass, and the mass, respectively, for the j^{th} node on the k^{th} body. The *structural mass matrix* for the k^{th} body $M_s(k)$ is the block diagnal matrix

$$M_s(k) = \text{diag}\Big\{M_s(j_k)\Big\} \in \Re^{6n_s(k)\times 6n_s(k)} \tag{2}$$

The *structural stiffness matrix* is denoted $K_s(k) \in \Re^{6n_s(k)\times 6n_s(k)}$. For a 3-vector x, there is a corresponding cross product matrix $\tilde{x}$. Both $M_s(k)$ and $K_s(k)$ are typically generated using finite element analysis.

As in Figure 1, the bodies in the serial chain are numbered in increasing order from tip to base. The

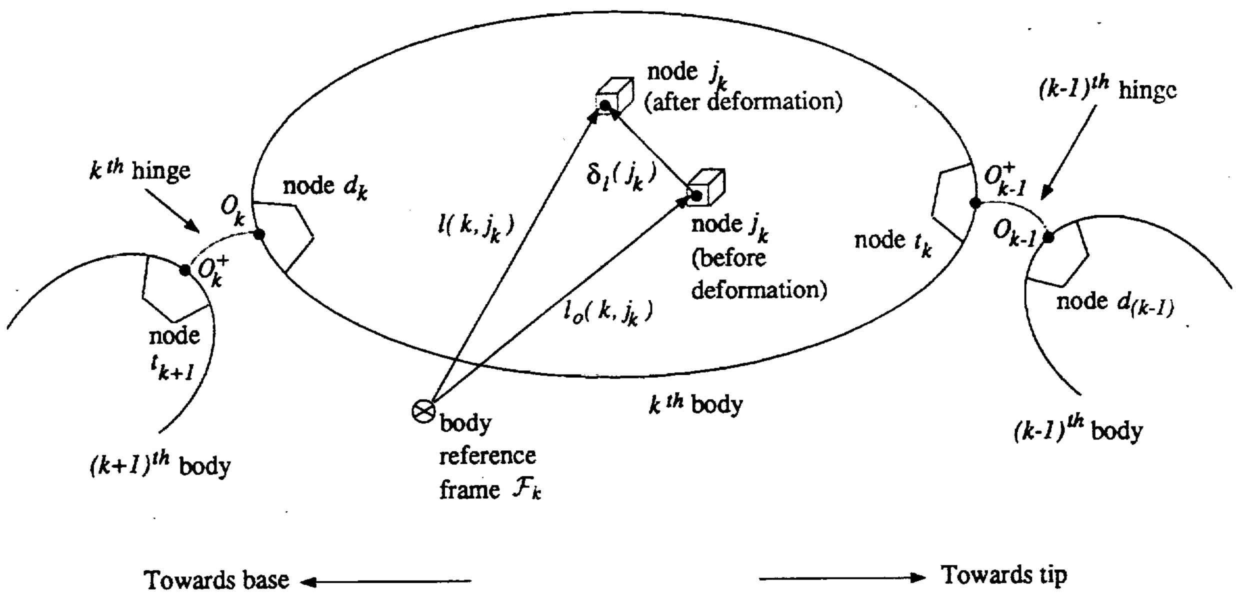

Figure 1: Illustration of links and hinges in a flexible serial multi-body system

terms *inboard (outboard)* denote the direction along the serial chain towards (away from) the base body. The k^{th} body is attached on the inboard side to the $(k+1)^{th}$ body by the k^{th} hinge, and on the outboard side to the $(k-1)^{th}$ body by the $(k-1)^{th}$ hinge. On the k^{th} body, the node to which the outboard hinge (the $(k-1)^{th}$ hinge) is attached is node t_k, while the node to which the inboard hinge (the k^{th} hinge) is attached is node d_k. The k^{th} hinge couples nodes d_k and t_{k+1}. Attached to each of these nodes are the k^{th} hinge reference frames $\mathcal{O}_k$ and $\mathcal{O}^+_k$ respectively. The number of degrees of freedom (dofs) for the k^{th} hinge is $n_r(k)$. The vector of configuration variables for the k^{th} hinge is $\theta(k) \in \Re^{n_r(k)}$, while the hinge's vector of generalized speeds is $\beta(k) \in \Re^{n_r(k)}$. In general, when there are nonholonomic hinge constraints, the dimensionality of $\beta(k)$ may be less than that of $\theta(k)$. For 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 quasi-coordinates simplifies the dynamical equations of motion, and there may be a better choice for $\beta(k)$. The relative spatial velocity $\Delta_V(k)$ across the hinge is $H^*(k)\beta(k)$, where $H^*(k)$ is the joint map matrix for the k^{th} hinge.

A set of $n_m(k)$ assumed modes is chosen for the k^{th} body. Let $\Pi_r^j(k) \in \Re^6$ be the *modal spatial displacement vector* at the j_k^{th} node for the r^{th} mode. The *modal spatial displacement influence vector* $\Pi^j(k) \in \Re^{6\times n_m(k)}$ for the j_k^{th} node and the *modal matrix* $\Pi(k) \in \Re^{6n_s(k)\times n_m(k)}$ for the k^{th} body are

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

The r^{th} column of $\Pi(k)$ is $\Pi_r(k)$, which defines the mode shape for the r^{th} assumed mode for the k^{th} body. Let $\eta(k) \in \Re^{n_m(k)}$ be the vector of modal deformation variables for the k^{th} body. The spatial deformation of node j_k and the spatial deformation field $u(k)$ for the k^{th} body are

$$u(j_k) = \Pi^j(k)\eta(k) \quad \text{and} \quad u(k) = \Pi(k)\eta(k) \tag{3}$$

Note that for cantilever modes

$$\Pi_r^d(k) = 0 \quad \text{and} \quad r = 1 \cdots n_m(k) \tag{4}$$

The vector of *generalized configuration variables* $\vartheta(k)$ and the vector of *generalized speeds* $\chi(k)$ for the k^{th} body are

$$\vartheta(k) \triangleq \begin{pmatrix} \eta(k) \\ \theta(k) \end{pmatrix} \in \Re^{\mathcal{N}(k)} \quad \text{and} \quad \chi(k) \triangleq \begin{pmatrix} \dot\eta(k) \\ \beta(k) \end{pmatrix} \in \Re^{\mathcal{N}(k)} \tag{5}$$

where $\mathcal{N}(k) \triangleq n_m(k) + n_r(k)$. The overall vectors of *generalized configuration variables* ϑ and *generalized speeds* χ for the serial multibody system are

$$\vartheta \triangleq \text{col}\left\{\vartheta(k)\right\} \in \Re^{\mathcal{N}} \quad \text{and} \quad \chi \triangleq \text{col}\left\{\chi(k)\right\} \in \Re^{\mathcal{N}} \tag{6}$$

where $\mathcal{N} \triangleq \sum_{k=1}^{N} \mathcal{N}(k)$. The number of overall degrees of freedom for the multibody system is $\mathcal{N}$. 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 relate the *generalized accelerations* $\dot\chi$ and *generalized forces* $T \in \Re^{\mathcal{N}}$. The *inverse dynamics* problem is to compute the generalized forces T for a prescribed set of *generalized accelerations* $\dot\chi$. Conversely, the *forward dynamics* problem is to compute the generalized accelerations $\dot\chi$ from the generalized forces T.

2.1 Recursive Propagation of Velocities

Let $V(k)$ be the spatial velocity of the k^{th} body reference frame $\mathcal{F}_k$, i.e.,

$$V(k) = \begin{pmatrix} \omega(k) \\ v(k) \end{pmatrix} \in \Re^6$$

where $\omega(k)$ and $v(k)$ are the angular and linear velocities respectively of $\mathcal{F}_k$. The spatial velocity $V_s(t_{k+1}) \in \Re^6$ of node t_{k+1} (on the inboard side of the k^{th} 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 $\dot\eta(k+1)$:

$$V_s(t_{k+1}) \quad = \quad \phi^*(k+1, t_{k+1})V(k+1) + \Pi^t(k+1)\dot\eta(k+1) \tag{7}$$

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

$$\phi(x,y) = \begin{pmatrix} I & \tilde{l}(x,y) \\ 0 & I \end{pmatrix} \tag{8}$$

where $l(x,y) \in \Re^3$ is the vector between the points x and y. Note the group property

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

for arbitrary points x, y and z. As in Eq. (7), and all through this paper, the index k will be used to refer to both the k^{th} body as well as to the k^{th} body reference frame $\mathcal{F}_k$ with the specific usage coming from the context. 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 $V(\mathcal{O}_k^+)$ of frame $\mathcal{O}_k^+$ (on the inboard side of the k^{th} hinge) is related to $V_s(t_{k+1})$ by

$$V(\mathcal{O}_k^+) = \phi^*(t_{k+1},\mathcal{O}_k)V_s(t_{k+1}) \tag{9}$$

The relative spatial velocity $\Delta_V(k)$ across the k^{th} hinge is $H^*(k)\beta(k)$, and so the spatial velocity $V(\mathcal{O}_k)$ of frame $\mathcal{O}_k$ on the outboard side of the k^{th} hinge is

$$V(\mathcal{O}_k) = V(\mathcal{O}_k^+) + H^*(k)\beta(k) \tag{10}$$

The spatial velocity $V(k)$ of the k^{th} body reference frame is

$$V(k) = \phi^*(\mathcal{O}_k,k)V(\mathcal{O}_k) - \mathring{u}(d_k) = \phi^*(\mathcal{O}_k,k)\left[V(\mathcal{O}_k) - \Pi^d(k)\dot{\eta}(k)\right] \tag{11}$$

Eq. (7), Eq. (9), Eq. (10) and Eq. (11) together imply

$$\begin{aligned} V(k) \;\; = \;\; & \phi^*(k+1,k)V(k+1) + \phi^*(t_{k+1},k)\Pi^t(k+1)\dot{\eta}(k+1) \\ & +\phi^*(\mathcal{O}_k,k)\left[H^*(k)\beta(k) - \Pi^d(k)\dot{\eta}(k)\right] \end{aligned} \tag{12}$$

Thus, with $\overline{\mathcal{N}}(k) \triangleq n_m(k) + 6$ and Eq. (12), the *modal spatial velocity* $V_m(k) \in \Re^{\overline{\mathcal{N}}(k)}$ for the k^{th} body is

$$V_m(k) \triangleq \begin{pmatrix} \dot{\eta}(k) \\ V(k) \end{pmatrix} = \Phi^*(k+1,k)V_m(k+1) + \mathcal{H}^*(k)\chi(k) \in \Re^{\overline{\mathcal{N}}(k)} \tag{13}$$

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

$$\Phi(k+1,k) \;\; \triangleq \;\; \begin{pmatrix} 0 & [\Pi^t(k+1)]^*\phi(t_{k+1},k) \\ 0 & \phi(k+1,k) \end{pmatrix} \in \Re^{\overline{\mathcal{N}}(k+1)\times\overline{\mathcal{N}}(k)} \tag{14}$$

$$\mathcal{H}(k) \;\; \triangleq \;\; \begin{pmatrix} I & -[\Pi^d_{\mathcal{F}}(k)]^* \\ 0 & H_{\mathcal{F}}(k) \end{pmatrix} \in \Re^{\mathcal{N}(k)\times\overline{\mathcal{N}}(k)} \tag{15}$$

where

$$H_{\mathcal{F}}(k) \triangleq H(k)\phi(\mathcal{O}_k,k) \in \Re^{n_r(k)\times 6}, \quad \text{and} \quad \Pi^d_{\mathcal{F}}(k) \triangleq \phi^*(\mathcal{O}_k,k)\Pi^d(k) \in \Re^{6\times\mathcal{N}(k)}$$

Note that

$$\Phi(k+1,k) = \mathcal{A}(k+1)\mathcal{B}(k+1,k) \tag{16}$$

where

$$\mathcal{A}(k) \triangleq \begin{pmatrix} [\Pi^t(k)]^* \\ \phi(k,t_k) \end{pmatrix} \in \Re^{\overline{\mathcal{N}}(k)\times 6} \quad \text{and} \quad \mathcal{B}(k+1,k) \triangleq [0, \;\; \phi(t_{k+1},k)] \in \Re^{6\times\overline{\mathcal{N}}(k)} \tag{17}$$

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

$$\mathcal{H}(k) = \begin{pmatrix} \mathcal{H}_f(k) \\ \mathcal{H}_r(k) \end{pmatrix} \in \Re^{\mathcal{N}(k) \times \overline{\mathcal{N}}(k)} \tag{18}$$

where

$$\mathcal{H}_f(k) \triangleq \left[I, \quad -[\Pi^d_{\mathcal{F}}(k)]^*\right] \in \Re^{n_m(k) \times \overline{\mathcal{N}}(k)} \quad \text{and} \quad \mathcal{H}_r(k) \triangleq [0, \quad H(k)\phi(\mathcal{O}_k, k)] \in \Re^{n_r(k) \times \overline{\mathcal{N}}(k)} \tag{19}$$

With $\overline{\mathcal{N}} = \sum_{k=1}^{N} \overline{\mathcal{N}}(k)$, the spatial operator $\mathcal{E}_\Phi$ is defined as

$$\mathcal{E}_\Phi \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \Phi(2,1) & 0 & \dots & 0 & 0 \\ 0 & \Phi(3,2) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \Phi(N, N-1) & 0 \end{pmatrix} \in \Re^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}} \tag{20}$$

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

$$\Phi \quad \triangleq \quad [I - \mathcal{E}_\Phi]^{-1} = I + \mathcal{E}_\Phi + \cdots + \mathcal{E}_\Phi^{N-1} = \begin{pmatrix} I & 0 & \dots & 0 \\ \Phi(2,1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(N,1) & \Phi(N,2) & \dots & I \end{pmatrix} \in \Re^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}} \tag{21}$$

where

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

Also define the spatial operator $\mathcal{H} \triangleq \text{diag}\left\{\mathcal{H}(k)\right\} \in \Re^{\mathcal{N} \times \overline{\mathcal{N}}}$. It follows that

$$V_m = \Phi^* \mathcal{H}^* \chi \tag{22}$$

where $V_m \triangleq \text{col}\left\{V_m(k)\right\} \in \Re^{\overline{\mathcal{N}}}$.

2.2 Modal Mass Matrix for a Single Body

An expression for the modal mass matrix of the k^{th} body is derived. Denote by $V_s(j_k) \in \Re^6$ the spatial velocity of node j_k. $V_s(k) \triangleq \text{col}\left\{V_s(j_k)\right\} \in \Re^{6n_s(k)}$ is the vector of all nodal spatial velocities for the k^{th} body. It follows (as in Eq. (7)) that

$$V_s(k) = B^*(k)V(k) + \mathring{u}(k) = [\Pi(k), \quad B^*(k)]V_m(k) \tag{23}$$

where

$$B(k) \triangleq [\phi(k, 1_k), \phi(k, 2_k), \cdots, \phi(k, n_s(k))] \in \Re^{6 \times 6n_s(k)} \tag{24}$$

Since $M_s(k)$ is the *structural mass matrix* of the k^{th} body, the kinetic energy of the k^{th} body is

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

where

$$M_m(k) \quad \triangleq \quad \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} M_s(k)[\Pi(k), \quad B^*(k)] = \begin{pmatrix} M_m^{ff}(k) & M_m^{fr}(k) \\ M_m^{rf}(k) & M_m^{rr}(k) \end{pmatrix} \in \Re^{\overline{\mathcal{N}}(k) \times \overline{\mathcal{N}}(k)} \tag{25}$$

Corresponding to the generalized speed vector $\chi(k)$, $M_m(k)$ is the *modal mass matrix* of the k^{th} body. In the block partitioning in Eq. (25), the superscripts f and r denote the *flexible* and *rigid* blocks respectively. Thus $M_m^{ff}(k)$ represents the flex/flex coupling block, while $M_m^{fr}(k)$ the flex/rigid coupling block, of $M_m(k)$. Note that $M_m^{rr}(k)$ is precisely the rigid body spatial inertia of the k^{th} 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).

Since 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. (25) it follows that while the block $\mathcal{M}_m^{ff}(k)$ is deformation independent, both the blocks $M_m^{fr}(k)$ and $M_m^{rr}(k)$ are deformation dependent. The detailed expression for the modal mass matrix can be defined using *modal integrals* which 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 k^{th} body can be found in [6]. Often the deformation dependent parts of the modal mass matrix are ignored, and free-free eigen–modes 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 block diagonal.

2.3 Recursive Propagation of Accelerations

Differentiation of the velocity recursion equation, Eq. (13), results in the following recursive expression for the *modal spatial acceleration* $\alpha_m(k) \in \Re^{\mathcal{N}(k)}$ for the k^{th} body:

$$\alpha_m(k) \triangleq \dot{V}_m(k) = \begin{pmatrix} \ddot{\eta}(k) \\ \alpha(k) \end{pmatrix} = \Phi^*(k+1,k)\alpha_m(k+1) + \mathcal{H}^*(k)\dot{\chi}(k) + a_m(k) \qquad \textbf{(26)}$$

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

$$a_m(k) = \frac{\mathrm{d}\Phi^*(k+1,k)}{\mathrm{d}t} V_m(k+1) + \frac{\mathrm{d}\mathcal{H}^*(k)}{\mathrm{d}t}\chi(k) \qquad \textbf{(27)}$$

The detailed expressions for $a_m(k)$ can be found in [6]. Defining $a_m = \mathrm{col}\{a_m(k)\} \in \Re^{\mathcal{N}}$ and $\alpha_m = \mathrm{col}\{\alpha_m(k)\} \in \Re^{\mathcal{N}}$, Eq. (26) can be reexpressed using spatial operators in the form

$$\alpha_m = \Phi^*(\mathcal{H}^*\dot{\chi} + a_m) \qquad \textbf{(28)}$$

The vector of spatial accelerations of all the nodes for the k^{th} body, $\alpha_s(k) \triangleq \mathrm{col}\{\alpha_s(j_k)\}$ $\in \Re^{6n_s(k)}$, is obtained by differentiating Eq. (23):

$$\alpha_s(k) = \dot{V}_s(k) = [\Pi(k), \;\; B^*(k)]\alpha_m(k) + a(k) \qquad \textbf{(29)}$$

where

$$a(k) \triangleq \mathrm{col}\{a(j_k)\} = \frac{\mathrm{d}[\Pi(k), \;\; B^*(k)]}{\mathrm{d}t} V_m(k) \in \Re^{6n_s(k)} \qquad \textbf{(30)}$$

2.4 Recursive Propagation of Forces

The equations of motion for the k^{th} body are now developed. Let $f(k-1) \in \Re^6$ denote the effective spatial force of interaction, referred to frame $\mathcal{F}_{k-1}$, between the k^{th} 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 k^{th} body and node d_{k-1} on the $(k-1)^{th}$ body. With $f_s(j_k) \in \Re^6$ denoting the spatial force at a node j_k, the force balance equation for node t_k is

$$f_s(t_k) \;\; = \;\; \phi(t_k, k-1)f(k-1) + M_s(t_k)\alpha_s(t_k) + b(t_k) + f_K(t_k) \qquad \textbf{(31)}$$

For all nodes other than node t_k on the k^{th} body, the force balance equation is

$$f_s(j_k) = M_s(j_k)\alpha_s(j_k) + b(j_k) + f_K(j_k) \tag{32}$$

In the above, $f_K(k) = K_s(k)u(k) \in \Re^{6n_s(k)}$ is the vector of spatial elastic strain forces for the nodes on the k^{th} body, while $b(j_k) \in \Re^6$ is the spatial gyroscopic force for node j_k and is given by

$$b(j_k) = \begin{pmatrix} \tilde{\omega}(j_k)\mathcal{J}(j_k)\omega(j_k) \\ m(j_k)\tilde{\omega}(j_k)\tilde{\omega}(j_k)p(j_k) \end{pmatrix} \in \Re^6 \tag{33}$$

where $\omega(j_k) \in \Re^3$ denotes the angular velocity of node j_k. Define

$$C(k,k-1) \triangleq \begin{pmatrix} 0 \\ \vdots \\ \phi(t_k,k-1) \\ \vdots \\ 0 \end{pmatrix} \in \Re^{6n_s(k)\times 6} \quad \text{and} \quad b(k) \triangleq \text{col}\Big\{b(j_k)\Big\} \in \Re^{6n_s(k)} \tag{34}$$

Eq. (31) and Eq. (32) imply

$$f_s(k) \;=\; C(k,k-1)f(k-1) + M_s(k)\alpha_s(k) + b(k) + K_s(k)u(k) \tag{35}$$

where $f_s(k) \triangleq \text{col}\Big\{f_s(j_k)\Big\} \in \Re^{6n_s(k)}$. Noting that

$$f(k) = B(k)f_s(k) \tag{36}$$

and using the principle of virtual work, it follows from Eq. (23) that the *modal spatial forces* $f_m(k) \in \Re^{\overline{\mathcal{N}}(k)}$ for the k^{th} body are

$$f_m(k) \;\triangleq\; \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} f_s(k) = \begin{pmatrix} \Pi^*(k)f_s(k) \\ f(k) \end{pmatrix} \tag{37}$$

Premultiplying Eq. (35) by $\begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix}$ and using Eq. (25), Eq. (29), and Eq. (37) leads to the following recursive relationship for the modal spatial forces:

$$f_m(k) \;=\; \Phi(k,k-1)f_m(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \tag{38}$$

where

$$b_m(k) \triangleq \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} [b(k) + M_s(k)a(k)] \in \Re^{\overline{\mathcal{N}}(k)} \tag{39}$$

and the *modal stiffness matrix*

$$K_m(k) \triangleq \begin{pmatrix} \Pi^*(k)K_s(k)\Pi(k) & 0 \\ 0 & 0 \end{pmatrix} \in \Re^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \tag{40}$$

The expression for $K_m(k)$ in Eq. (40) uses the fact that the columns of $B^*(k)$ are the *deformation dependent* rigid body modes for the k^{th} body, and hence they do not contribute to its elastic strain energy. When a deformation dependent structural stiffness matrix $K_s(k)$ is used,

$$K_s(k)B^*(k) = 0 \tag{41}$$

However, common practice (followed in this paper) uses a constant, deformation–independent structural stiffness matrix. This leads to the apparently anomalous situation wherein Eq. (41) does not hold exactly. All these fictitious extra terms on the left–hand side of Eq. (41) are commonly ignored.

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 [6]. From Eq. (38), the operator expression for the modal spatial forces $f_m \triangleq \text{col}\{f_m(k)\} \in \Re^{\overline{\mathcal{N}}}$ for all the bodies in the chain is

$$f_m = \Phi(M_m \alpha_m + b_m + K_m \vartheta) \tag{42}$$

where

$$\mathcal{M}_m \triangleq \text{diag}\{M_m(k)\} \in \Re^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}},\ K_m \triangleq \text{diag}\{K_m(k)\} \in \Re^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}},\ \text{and}\ b_m \triangleq \text{col}\{b_m(k)\} \in \Re^{\overline{\mathcal{N}}}$$

From the principle of virtual work, the *generalized forces* vector $T \in \Re^{\mathcal{N}}$ for the multibody system is

$$T = \mathcal{H} f_m \tag{43}$$

2.5 Operator Expression for the System Mass Matrix

Collection of the operator expressions in Eq. (22), Eq. (28), Eq. (42) and Eq. (43) leads to:

$$\begin{aligned}
V_m &= \Phi^* \mathcal{H}^* \chi \\
\alpha_m &= \Phi^* (\mathcal{H}^* \dot{\chi} + a_m) \\
f_m &= \Phi(M_m \alpha_m + b_m + K_m \vartheta) = \Phi M_m \Phi^* \mathcal{H}^* \dot{\chi} + \Phi(M_m \Phi^* a_m + b_m + K_m \vartheta) \\
T &= \mathcal{H} f_m = \mathcal{H} \Phi M_m \Phi^* \mathcal{H}^* \dot{\chi} + \mathcal{H} \Phi (M_m \Phi^* a_m + b_m) \\
&= \mathcal{M} \dot{\chi} + \mathcal{C}
\end{aligned} \tag{44}$$

where

$$\mathcal{M} \triangleq \mathcal{H} \Phi M_m \Phi^* \mathcal{H}^* \in \Re^{\mathcal{N} \times \mathcal{N}} \quad \text{and} \quad \mathcal{C} \triangleq \mathcal{H} \Phi (M_m \Phi^* a_m + b_m + K_m \vartheta) \in \Re^{\mathcal{N}} \tag{45}$$

Here $\mathcal{M}$ is the system mass matrix. The expression $\mathcal{H} \Phi M_m \Phi^* \mathcal{H}^*$ is referred to as the *Newton-Euler Operator Factorization* of the mass matrix. The term $\mathcal{C}$ is the vector of Coriolis, centrifugal, and elastic forces for the system.

The operator expressions for $\mathcal{M}$ and $\mathcal{C}$ are identical in form to those for rigid multibody systems (see [1, 7]). This similarity is extremely useful in the extension of recursive algorithms from rigid multibody systems to flexible multibody systems.

3. 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 (a) computing the system mass matrix $\mathcal{M}$, (b) computing the bias vector $\mathcal{C}$, and (c) solving the linear matrix equation for $\dot{\chi}$:

$$\mathcal{M} \dot{\chi} = T - \mathcal{C} \tag{46}$$

Section 4 describes the recursive articulated body forward dynamics algorithm that does not require the explicit computation of either $\mathcal{M}$ or $\mathcal{C}$.

Lemma 3.1: Define the *composite body inertias* $R(k) \in \Re^{\overline{\mathcal{N}}(k) \times \overline{\mathcal{N}}(k)}$ recursively for all the bodies in the serial chain as follows:

$$\left\{ \begin{aligned}
& R(0) = 0 \\
& \textbf{for k} = \mathbf{1 \cdots N} \\
& \quad R(k) = \Phi(k, k-1) R(k-1) \Phi^*(k, k-1) + M_m(k) \\
& \textbf{end loop}
\end{aligned} \right. \tag{47}$$

Also define $R \stackrel{\triangle}{=} \text{diag}\left\{R(k)\right\} \in \Re^{\mathcal{N}\times\mathcal{N}}$. Then, observe the following spatial operator decomposition where $\tilde{\Phi} \stackrel{\triangle}{=} \Phi - I$:

$$\Phi M_m \Phi^* \;=\; R + \tilde{\Phi} R + R\tilde{\Phi}^* \tag{48}$$

Physically, $R(k)$ is the modal mass matrix of the composite body formed from all the bodies outboard of the k^{th} hinge by freezing all their (deformation plus hinge) degrees of freedom. It follows from Eq. (45) and Lemma 3.1 that

$$\mathcal{M} = \mathcal{H}\Phi M_m \Phi^* \mathcal{H}^* = \mathcal{H}R\mathcal{H}^* + \mathcal{H}\tilde{\Phi}R\mathcal{H}^* + \mathcal{H}R\tilde{\Phi}^*\mathcal{H}^* \tag{49}$$

Note that the three terms on the right of Eq. (49) are block diagonal, block lower triangular and block upper triangular respectively. The algorithm for computing the mass matrix $\mathcal{M}$ computes these terms recursively. The main recursion proceeds from tip to base, and computes the blocks along the diagonal of $\mathcal{M}$. As each such diagonal element is computed, a new recursion to compute the off-diagonal elements is spawned. Its structure is similar to that of the composite body algorithm for computing the mass matrix of rigid multibody systems (see [8,9]), and is as follows:

$$\left\{\begin{array}{l} \quad R(0) = 0 \\ \textbf{for k} = \mathbf{1 \cdots N} \\ \quad \begin{array}{rcl} R(k) &=& \Phi(k,k-1)R(k-1)\Phi^*(k,k-1) + M_m(k) \\ &=& \mathcal{A}(k)\phi(t_k,k-1)R^{rr}(k-1)\phi^*(t_k,k-1)\mathcal{A}^*(k) + M_m(k) \\ X(k) &=& R(k)\mathcal{H}^*(k) \\ \mathcal{M}_s(k,k) &=& \mathcal{H}(k)X(k) \\ && \left\{\begin{array}{l} \textbf{for j} = \mathbf{(k+1)\cdots N} \\ \quad \begin{array}{rcl} X(j) &=& \Phi(j,j-1)X(j-1) = \mathcal{A}(j)\phi(t_j,j-1)X^r(j-1) \\ \mathcal{M}(j,k) &=& \mathcal{M}^*(k,j) = \mathcal{H}(j)X(j) \end{array} \\ \textbf{end loop} \end{array}\right. \end{array} \\ \textbf{end loop} \end{array}\right. \tag{50}$$

The structure of the above algorithm for computing the mass matrix closely resembles the composite rigid body algorithm for computing the mass matrix of rigid multibody systems [8,9]. Like the latter, it is also highly efficient. Additional computational simplifications of the algorithm arising from the sparsity of both $\mathcal{H}_f(k)$ and $\mathcal{H}_r(k)$ are easy to incorporate.

4. Factorization and Inversion of the Mass Matrix

An operator factorization of the system mass matrix $\mathcal{M}$, referred to as the *Innovations Operator Factorization*, is derived. This factorization is an alternative to the Newton–Euler factorization in Eq. (45). In contrast with the latter, the factors in the Innovations factorization are square and invertible. Operator expressions for the inverse of these factors lead to an operator expression for the inverse of the mass matrix. Use of further operator identities results in the recursive articulated body forward dynamics algorithm in Section 5. The operator factorization and inversion results here closely resemble those for rigid multibody systems (see [1]).

The following recursive algorithm defines some required articulated body quantities. This algorithm has the structure of the Riccati equation of Kalman filtering theory [9]:

$$
\left\{
\begin{array}{rcl}
 & & P^{+}(0) = 0 \\
\textbf{for k} & = & \mathbf{1 \cdots N} \\
P(k) & = & \Phi(k,k-1)P^{+}(k-1)\Phi^{*}(k,k-1) + M_m(k) \in \Re^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
D(k) & = & \mathcal{H}(k)P(k)\mathcal{H}^{*}(k) \in \Re^{\mathcal{N}(k)\times\mathcal{N}(k)} \\
G(k) & = & P(k)\mathcal{H}^{*}(k)D^{-1}(k) \in \Re^{\overline{\mathcal{N}}(k)\times\mathcal{N}(k)} \\
K(k+1,k) & = & \Phi(k+1,k)G(k) \in \Re^{\overline{\mathcal{N}}(k)\times\mathcal{N}(k)} \\
\overline{\tau}(k) & = & I - G(k)\mathcal{H}(k) \in \Re^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
P^{+}(k) & = & \overline{\tau}(k)P(k) \in \Re^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
\Psi(k+1,k) & = & \Phi(k+1,k)\overline{\tau}(k) \in \Re^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
\textbf{end loop} & &
\end{array}
\right. \tag{51}
$$

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

$$
\begin{array}{rcl}
D & \triangleq & \mathcal{H}P\mathcal{H}^{*} = \text{diag}\Big\{D(k)\Big\} \in \Re^{\mathcal{N}\times\mathcal{N}} \\
G & \triangleq & P\mathcal{H}^{*}D^{-1} = \text{diag}\Big\{G(k)\Big\} \in \Re^{\overline{\mathcal{N}}\times\mathcal{N}} \\
K & \triangleq & \mathcal{E}_{\Phi}G \in \Re^{\overline{\mathcal{N}}\times\mathcal{N}} \\
\overline{\tau} & \triangleq & I - G\mathcal{H} = \text{diag}\Big\{\overline{\tau}(k)\Big\} \in \Re^{\overline{\mathcal{N}}\times\overline{\mathcal{N}}} \\
\mathcal{E}_{\Psi} & \triangleq & \mathcal{E}_{\Phi}\overline{\tau} \in \Re^{\overline{\mathcal{N}}\times\overline{\mathcal{N}}}
\end{array} \tag{52}
$$

The only nonzero block elements of K and $\mathcal{E}_{\Psi}$ are the elements $K(k+1,k)$ and $\Psi(k+1,k)$ respectively along the first sub-diagonal. The spatial operator G is formed by a set of spatial Kalman gains [9] for a spatially recursive Kalman filter.

As in the case for $\mathcal{E}_{\Phi}$, $\mathcal{E}_{\Psi}$ is nilpotent, so the operator Ψ can be defined as

$$
\Psi \triangleq (I - \mathcal{E}_{\Psi})^{-1} = \begin{pmatrix} I & 0 & \dots & 0 \\ \Psi(2,1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Psi(N,1) & \Psi(N,2) & \dots & I \end{pmatrix} \in \Re^{\overline{\mathcal{N}}\times\overline{\mathcal{N}}} \tag{53}
$$

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 respectively 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 lemmas established in [1].

Lemma 4.1:

$$
\mathcal{M} = [I + \mathcal{H}\Phi K]D[I + \mathcal{H}\Phi K]^{*} \tag{54}
$$

Note that the factor $[I + \mathcal{H}\Phi K] \in \Re^{\mathcal{N}\times\mathcal{N}}$ is square, block lower triangular and nonsingular, while D is a block diagonal matrix. This factorization may be regarded as a block LDL^* decomposition of $\mathcal{M}$. The following lemma gives the closed form operator expression for the inverse of the factor $[I + \mathcal{H}\Phi K]$.

Lemma 4.2:

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

Lemma 4.3:

$$\mathcal{M}^{-1} = [I - \mathcal{H}\Psi K]^* D^{-1} [I - \mathcal{H}\Psi K] \tag{56}$$

Once again, note that the factor $[I - \mathcal{H}\Psi K]$ is square, block lower triangular and nonsingular and so Lemma 4.3 may be regarded as providing a block LDL^* decomposition of $\mathcal{M}^{-1}$. This decomposition however is model-based, in the sense that the physical model of the system is used to conduct computations. This means that every step in the decomposition has a corresponding physical interpretation which adds a substantial amount of insight into the decomposition.

5. Articulated Body Forward Dynamics Algorithm

The operator-based mass matrix inverse leads to a recursive forward dynamics algorithm. The structure of this algorithm is completely identical in form to the articulated body algorithm for serial *rigid* multibody systems. Its structure is that of a Kalman filter and a Bryson-Frazier smoother [9].

The following lemma, established in [1], describes the operator expression for the generalized accelerations $\dot{\chi}$ in terms of the generalized forces T.

Lemma 5.1:

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

As in the case of rigid multibody systems [1, 10], the direct recursive implementation of Eq. (57) leads to the following recursive forward dynamics algorithm:

$$\left\{\begin{array}{rcl}
 & z^+(0) = 0 & \\
\textbf{for k} & = & \textbf{1} \cdots \textbf{n} \\
z(k) & = & \Phi(k, k-1) z^+(k-1) + P(k) a_m(k) + b_m(k) + K_m(k)\vartheta(k) \\
\epsilon(k) & = & T(k) - \mathcal{H}(k) z(k) \\
\nu(k) & = & D^{-1}(k)\epsilon(k) \\
z^+(k) & = & z(k) + G(k)\epsilon(k) \\
\textbf{end loop} & &
\end{array}\right.$$

$$\left\{\begin{array}{rcl}
 & \alpha_m(n+1) = 0 & \\
\textbf{for k} & = & \textbf{n} \cdots \textbf{1} \\
\alpha_m^+(k) & = & \Phi^*(k+1, k)\alpha_m(k+1) \\
\dot{\chi}(k) & = & \nu(k) - G^*(k)\alpha_m^+(k) \\
\alpha_m(k) & = & \alpha_m^+(k) + \mathcal{H}^*(k)\dot{\chi}(k) + a_m(k) \\
\textbf{end loop} & &
\end{array}\right. \tag{58}$$

All the degrees of freedom for each body are characterized by its joint map matrix $\mathcal{H}^*(.)$ and 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 dofs and the hinge degrees of freedom separately. These simplifications are described in the following sections.

Instead of giving detail, the conceptual approach to separating modal and hinge degrees of freedom is described. First, recall the velocity recursion equation in Eq. (13)

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

and the partitioned form of $\mathcal{H}(k)$ in Eq. (15)

$$\mathcal{H}(k) = \begin{pmatrix} \mathcal{H}_f(k) \\ \mathcal{H}_r(k) \end{pmatrix} \tag{60}$$

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

$$\begin{aligned} V_m(k') &= \Phi^*(k+1,k')V_m(k+1) + \mathcal{H}_f^*(k)\dot{\eta}(k) \\ V_m(k) &= \Phi^*(k',k)V_m(k') + \mathcal{H}_r^*(k)\beta(k) \end{aligned} \tag{61}$$

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 bodies. The first one has the same kinematical and mass/inertia properties as the real body and is associated with the flexible degrees of freedom. The second body is a fictitious body, and it 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 the new bodies being fictitious. The new $\mathcal{H}^*$ operator has the same number of columns but twice the number of rows as the original $\mathcal{H}^*$ operator. The new Φ operator has twice the number of rows as well as twice the number of columns as the original. An analysis similar to those of the previous sections leads to an operator expression similar to Eq. (57). This implies a recursive forward dynamics algorithm like Eq. (58). However each sweep in the algorithm now contains twice as many steps as the original algorithm. But since each step now processes a smaller number of degrees of freedom, the overall computational cost is reduced.

5.1 Simplified Articulated Body Forward Dynamics Algorithm

The complete recursive *articulated body forward dynamics algorithm* for a serial flexible multibody system follows from recursive implementation of Eq. (57). The algorithm has the following steps: (a) compute the articulated body quantities, (b) do a base-to-tip recursion for the modal spatial velocities $V_m(k)$ and the bias terms $a_m(k)$, $b_m(k)$, and (c) do a tip-to-base recursion followed by a base-to-tip recursion for the joint accelerations $\dot{\chi}$.

$$\left\{\begin{array}{rcl} & P_R^+(0) = 0 & \\ \textbf{for k} & = & \textbf{1} \cdots \textbf{N} \\ \Gamma(k) & = & \phi(t_k, k-1) P_R^+(k-1) \phi^*(t_k, k-1) \\ P(k) & = & \mathcal{A}(k)\Gamma(k)\mathcal{A}^*(k) + M_m(k) \\ D_f(k) & = & \mathcal{H}_f(k) P(k) \mathcal{H}_f^*(k) \\ \mu(k) & = & [P^{rf}(k), \quad P^{rr}(k)]\mathcal{H}_f^*(k) \\ g(k) & = & \mu(k) D_f^{-1}(k) \\ P_R(k) & = & P^{rr}(k) - g(k)\mu^*(k) \\ D_R(k) & = & H_{\mathcal{F}}(k) P_R(k) H_{\mathcal{F}}^*(k) \\ G_R(k) & = & P_R(k) H_{\mathcal{F}}^*(k) D_R^{-1}(k) \\ \overline{\tau}_R(k) & = & I - G_R(k) H_{\mathcal{F}}(k) \\ P_R^+(k) & = & \overline{\tau}_R(k) P_R(k) \\ \textbf{end loop} & & \end{array}\right. \tag{62}$$

$$\left\{\begin{array}{rcl} & z_R^+(0) = 0 & \\ \textbf{for k} & = & \textbf{1} \cdots \textbf{N} \\ z(k) & = & \begin{pmatrix} z_f(k) \\ z_r(k) \end{pmatrix} \\ & = & \mathcal{A}(k)\phi(t_k, k-1) z_R^+(k-1) + b_m(k) + K_m(k)\vartheta(k) \in \Re^{\overline{\mathcal{N}}(k)} \\ \epsilon_f(k) & = & T_f(k) - z_f(k) + [\Pi_{\mathcal{F}}^d(k)]^* z_r(k) \in \Re^{n_m(k)} \\ \nu_f(k) & = & D_f^{-1}(k)\epsilon_f(k) \in \Re^{n_m(k)} \\ & & \\ z_R(k) & = & z_r(k) + g(k)\epsilon_f(k) + P_R(k) a_{mR}(k) \in \Re^6 \\ \epsilon_R(k) & = & T_R(k) - H_{\mathcal{F}}(k) z_R(k) \in \Re^{n_r(k)} \\ \nu_R(k) & = & D_R^{-1}(k)\epsilon_R(k) \in \Re^{n_r(k)} \\ z_R^+(k) & = & z_R(k) + G_R(k)\epsilon_R(k) \in \Re^6 \\ \textbf{end loop} & & \end{array}\right.$$

$$\left\{\begin{array}{rcl} & \alpha_m(N+1) = 0 & \\ \textbf{for k} & = & \textbf{N} \cdots \textbf{1} \\ \alpha_R^+(k) & = & \phi^*(t_{k+1}, k)\mathcal{A}^*(k+1)\alpha_m(k+1) \in \Re^6 \\ \beta(k) & = & \nu_R(k) - G_R^*(k)\alpha_R^+(k) \in \Re^{n_r(k)} \\ \alpha_R(k) & = & \alpha_R^+(k) + H_{\mathcal{F}}^*(k)\beta(k) + a_{mR}(k) \in \Re^6 \\ \ddot{\eta}(k) & = & \nu_f(k) - g^*(k)\alpha_R(k) \in \Re^{n_m(k)} \\ \alpha_m(k) & = & \begin{pmatrix} \ddot{\eta}(k) \\ \alpha_R(k) - \Pi_{\mathcal{F}}^d(k)\ddot{\eta}(k) \end{pmatrix} \in \Re^{\overline{\mathcal{N}}(k)} \\ \textbf{end loop} & & \end{array}\right. \tag{63}$$

The recursion in Eq. (63) is obtained by carrying out simplifications of the recursions in Eq. (58) in the same manner as described in the previous section for the articulated body quantities.

In contrast with the composite body forward dynamics algorithm of Section 3, this algorithm does not explicitly compute either $\mathcal{M}$ or $\mathcal{C}$. This algorithm is similar to those for rigid multibody systems [1,11].

C.3

6. Computational Cost

The computational costs of the composite body and the articulated body forward dynamics algorithms are compared. For low-spin multibody systems, it has been suggested in [12] 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 than the full nonlinear models and do not require much of the data on modal integrals 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 [6].

6.1 Computational Cost of the Composite Body Forward Dynamics Algorithm

The composite body forward dynamics algorithm described in Section 3 is based on solving the linear matrix equation

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

The computational cost of this forward dynamics algorithm is as follows:

1. The cost of computing $R(k)$ for the k^{th} body by using the algorithm in Eq. (50) is $[48n_m(k) + 90]\mathrm{M} + [n_m^2(k) + \frac{97}{2}n_m(k) + 116]\mathrm{A}$.
2. The contribution of the k^{th} body to the cost of computing $\mathcal{M}$ (excluding cost of $R(k)$'s) using the algorithm in Eq. (50) is
$\{k[12n_m^2(k) + 34n_m(k) + 13]\}\ \mathrm{M} + \{k[11n_m^2(k) + 24n_m(k) + 13]\}\ \mathrm{A}$.
3. Setting the generalized accelerations $\dot{\chi} = 0$, the vector C can be obtained by using an inverse dynamics algorithm for computing the generalized forces T. The contribution of the k^{th} body to the computational cost for $C(k)$ is $\{2n_m^2(k) + 54n_m(k) + 206\}\ \mathrm{M} + \{2n_m^2(k) + 50n_m(k) + 143\}\ \mathrm{A}$.
4. The cost of computing $T - C$ is $\{\mathcal{N}\}\ \mathrm{A}$.
5. The cost of solving the linear equation in Eq. (46) for the accelerations $\dot{\chi}$ is
$\left\{\frac{1}{6}\mathcal{N}^3 + \frac{3}{2}\mathcal{N}^2 - \frac{2}{3}\mathcal{N}\right\}\mathrm{M} + \left\{\frac{1}{6}\mathcal{N}^3 + \mathcal{N}^2 - \frac{7}{6}\mathcal{N}\right\}\mathrm{A}$.

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

6.2 Computational Cost of the Articulated Body Forward Dynamics Algorithm

The articulated body forward dynamics algorithm is based on the recursions described in Eq. (62) and Eq. (63). Since the computations in Eq. (47) can be done 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. (62). The step involving the computation of $D^{-1}(k)$ can be carried out either by an explicit inversion of $D(k)$ with $O(n_m^3(k))$ cost, or by the indirect procedure described in Eq. (62) with $O(n_m^2(k))$ cost. The first method is more efficient than the second one for $n_m(k) \leq 7$.
   - Cost of Eq. (62) for the k^{th} body, based on the explicit inversion of $D(k)$ (used when $n_m(k) \leq 7$), is
   $\left\{\frac{5}{6}n_m^3(k) + \frac{25}{2}n_m^2(k) + \frac{764}{3}n_m(k) + 180\right\}\mathrm{M} + \left\{\frac{5}{6}n_m^3(k) + \frac{21}{2}n_m^2(k) + \frac{548}{3}n_m(k) + 161\right\}\mathrm{A}$.
   - Cost of Eq. (62) for the k^{th} body based on the indirect computation of $D^{-1}(k)$ (used when $n_m(k) \geq 8$) is $\{12n_m^2(k) + 255n_m(k) + 572\}\ \mathrm{M} + \{13n_m^2(k) + 182n_m(k) + 445\}\ \mathrm{A}$.

2. The cost for the tip-to-base recursion sweep in Eq. (63) for the k^{th} body is $\{n_m^2(k) + 25n_m(k) + 49\}$ M + $\{n_m^2(k) + 24n_m(k) + 50\}$ A.

3. The cost for the base-to-tip recursion sweep in Eq. (63) for the k^{th} body is $\{18n_m(k) + 52\}$ M+ $\{19n_m(k) + 42\}$ 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.

The articulated body algorithm is more efficient than the composite body algorithm as the number of modes and bodies in the multibody system increases. Figure 2 contains a plot of the computa-

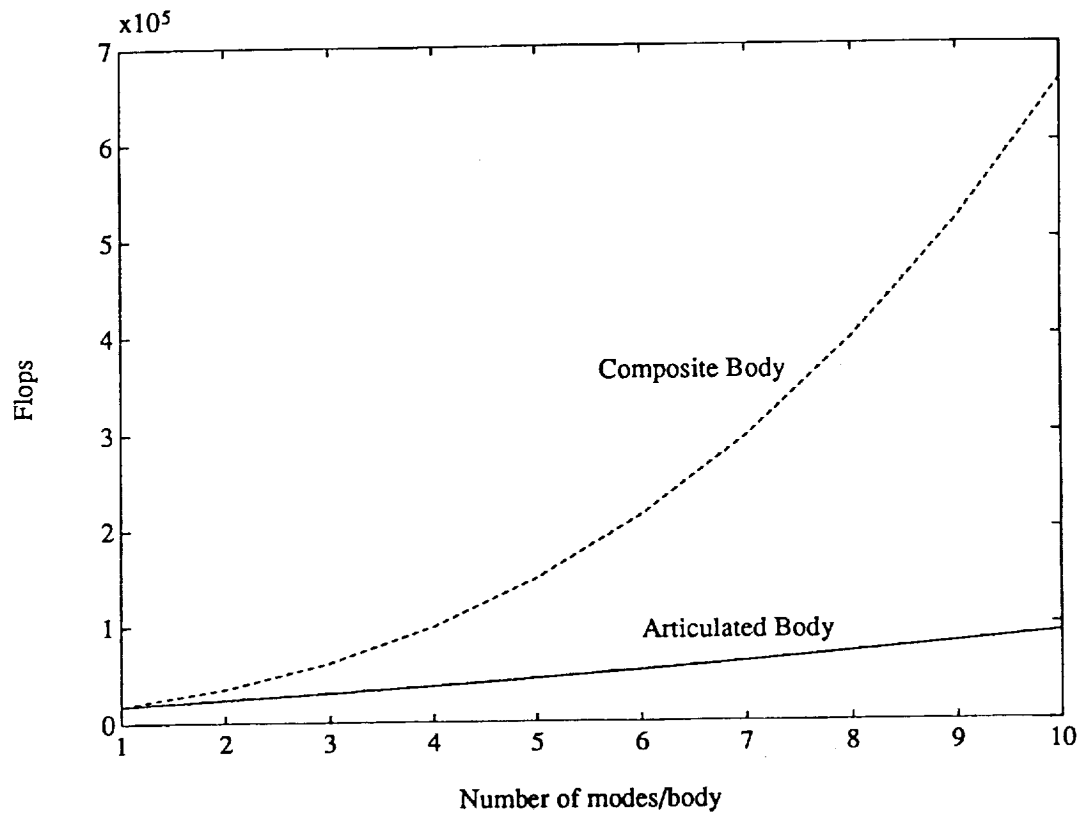

Figure 2: A comparison of computational costs for the forward dynamics algorithms for a flexible multibody serial chain system with 10 flexible bodies.

tional cost (in floating point operations) versus the number of modes per body for a serial chain with ten 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 10 modes per body. The divergence between the costs for the two algorithms becomes even more rapid as the number of bodies is increased.

7. Extensions to General Topology Flexible Multibody Systems

Extension to general tree and closed-chain systems is similar to methods given in prior results for rigid body configurations [7]. The key is that the operator description does not change as the topology changes. Extending the serial chain results of this paper to tree topologies takes the following steps:

1. For any outward 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 an inward tip(s) to base recursion, at each body, the recursion must be continued inward only after summing up contributions from each of the incoming branches of the body.

A closed-chain flexible multibody system can be regarded as a tree topology system with additional closure constraints [7].

8. Conclusions

This paper uses spatial operator methods to develop a new dynamics formulation and spatially recursive algorithms for flexible multibody systems. 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 unified approach is that it allows ideas and techniques for rigid multibody systems to be easily applied to flexible multibody systems. All of the computations are mechanized within a spatially recursive Kalman filtering and smoothing architecture. An extension of this algorithm to handle prescribed motion is described in reference [13].

The computational efficiency of the dynamics algorithms described in this paper makes it possible to implement real-time, high-fidelity, hardware-in-the-loop simulation of complex multibody systems such as spacecraft, robot manipulators, vehicles etc. Such simulations are essential during the design and testing of control and fault recovery algorithms. The articulated body forward dynamics algorithm is currently being used to simulate the dynamics of planetary spacecraft. One application is a spacecraft currently being assembled for a comet and asteroid rendezvous mission [14]. The multibody model for the spacecraft is of tree topology, and consists of a flexible central bus with 9 articulated appendages and 22 hinges' degrees of freedom. The simulation software provides a new capability for high speed simulation of the spacecraft. A real-time version has also been developed. Validation of this software was carried out by running independent simulations of the spacecraft using a standard flexible multibody simulation package [15]. Results from the two independent simulations show complete agreement.

9. Acknowledgement

The research described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Rodriguez, G., Kreutz-Delgado, K., and Jain, A., "A Spatial Operator Algebra for Manipulator Modeling and Control," *The International Journal of Robotics Research*, vol. 10, pp. 371–381, Aug. 1991.

[2] Jain, A. and Rodriguez, G., "Recursive Flexible Multibody System Dynamics Using Spatial Operators," *Journal of Guidance, Control and Dynamics*, vol. 15, pp. 1453–1466, Nov. 1992.

[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, pp. 293–314, 1988.

[4] Changizi, K. and Shabana, A.A., "A Recursive Formulation for the Dynamic Analysis of Open Loop Deformable Multibody Systems," *ASME Jl. of Applied Mechanics*, vol. 55, pp. 687–693, Sept. 1988.

[5] Keat, J.E., "Multibody System Order n Dynamics Formulation Based on Velocity Transform Method," *Journal of Guidance, Control and Dynamics*, vol. 13, March-April 1990.

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

[7] Rodriguez, G., Jain, A., and Kreutz-Delgado, K., "Spatial Operator Algebra for Multibody System Dynamics," *Journal of the Astronautical Sciences*, vol. 40, pp. 27–50, Jan.–March 1992.

[8] Walker, M.W. and Orin, D.E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 205–211, Sept. 1982.

[9] Rodriguez, G., "Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 624–639, Dec. 1987.

[10] Jain, A., "Unified Formulation of Dynamics for Serial Rigid Multibody Systems," *Journal of Guidance, Control and Dynamics*, vol. 14, pp. 531–542, May–June 1991.

[11] Featherstone, R., "The Calculation of Robot Dynamics using Articulated-Body Inertias," *The International Journal of Robotics Research*, vol. 2, pp. 13–30, Spring 1983.

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

[13] Jain, A. and Rodriguez, G., "Recursive Dynamics Algorithm for Multibody Systems with Prescribed Motion," *Journal of Guidance, Control and Dynamics*, 1992. In press.

[14] Bell, C.E., Bernard, D.E., and Rasmussen, R.D., "Attitude and Articulation Control for CRAF/Cassini," in *First ESA International Conference on Spacecraft Guidance, Navigation and Control Systems*, (Noordwijk, The Netherlands), June 1991.

[15] 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 Technical Paper 1219, NASA, Goddard Space Flight Center, May 1978.