Journal of Guidance, Control and Dynamics, vol. 15, pp. 1453-1466, Nov. 1992

Recursive Flexible Multibody System Dynamics using Spatial Operators

A. Jain* and G. Rodriguez†

Abstract

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.

1 Nomenclature

We use coordinate–free spatial notation (Refs. 3, 4) in this paper. A *spatial velocity* of a frame is a 6-dimensional quantity whose upper 3 elements are the angular velocity and whose lower 3 elements are the linear velocity. A *spatial force* is a 6-dimensional quantity whose upper 3 elements are a moment vector and whose lower 3 elements are a force vector.

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

*Member of Technical Staff, Nonlinear Science and Information Processing Group, Robotic Systems and Advanced Computer Technology Section, Jet Propulsion Laboratory/California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

†Section Staff, Robotic Systems and Advanced Computer Technology Section, Jet Propulsion Laboratory/California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

General Quantities:

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

$\dot{x}$ $= \frac{\boldsymbol{d}x}{\boldsymbol{d}t}$ - the time derivative of x with respect to an inertial frame

$\mathring{x}$ - the time derivative of x with respect to the body–fixed (rotating) frame

$\text{diag}\left\{x(k)\right\}$ - a block diagonal matrix whose k^{th} diagonal element is $x(k)$

$\text{col}\left\{x(k)\right\}$ - a column vector whose k^{th} element is $x(k)$

$l(x,y)$ $\in \mathbb{R}^3$ - the vector from point/frame x to point/framee y

$\phi(x,y)$ $= \begin{pmatrix} I & \tilde{l}(x,y) \\ 0 & I \end{pmatrix} \in \mathbb{R}^{6\times 6}$ - the spatial transformation operator which transforms spatial velocities and forces between points/frames x and y

Individual Body Nodal Data:

$n_s(k)$ - number of nodes on the k^{th} body

$\mathcal{F}_k$ - body reference frame with respect to which the deformation field for the k^{th} body is measured. The motion of this frame characterizes the motion of the k^{th} body as a rigid body.

j_k - j^{th} node on the k^{th} body

$l_0(k, j_k)$ $\in \mathbb{R}^3$ - vector from $\mathcal{F}_k$ to the location (before deformation) of the j^{th} node reference frame on the k^{th} body

$\delta_l(j_k)$ $\in \mathbb{R}^3$ - translational deformation of the j^{th} node on the k^{th} body

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

$\delta_\omega(j_k)$ $\in \mathbb{R}^3$ - deformation angular velocity of the j^{th} node on the k^{th} body with respect to the body frame $\mathcal{F}_k$

$\delta_v(j_k)$ $\in \mathbb{R}^3$ - deformation linear velocity of the j^{th} node on the k^{th} body with respect to the body frame $\mathcal{F}_k$

$u(j_k)$ $\in \mathbb{R}^6$ - the spatial displacement of node j_k. The translational component of $u(j_k)$ is $\delta_l(j_k)$, while its time derivative with respect to the body frame $\mathcal{F}_k$ is $\mathring{u}(j_k) = \begin{pmatrix} \delta_\omega(j_k) \\ \delta_v(j_k) \end{pmatrix}$

$\mathcal{J}(j_k)$ $\in \mathbb{R}^{3\times 3}$ - inertia tensor about the nodal reference frame for the j^{th} node on the k^{th} body

$p(j_k)$ $\in \mathbb{R}^3$ - vector from the nodal reference frame to the node center of mass for the j^{th} node on the k^{th} body

$m(j_k)$ - mass of the j^{th} node on the k^{th} body

$M_s(j_k)$ $= \begin{pmatrix} \mathcal{J}(j_k) & m(j_k)\tilde{p}(j_k) \\ -m(j_k)\tilde{p}(j_k) & m(j_k)I \end{pmatrix} \in \mathbb{R}^{6\times 6}$ - spatial inertia about the nodal reference frame for the j^{th} node on the k^{th} body

$M_s(k)$ $\text{diag}\left\{M_s(j_k)\right\} \in \mathbb{R}^{6n_s(k)\times 6n_s(k)}$ - structural mass matrix for the k^{th} body

$K_s(k)$ $\in \mathbb{R}^{6n_s(k)\times 6n_s(k)}$ - structural stiffness matrix for the k^{th} body

Individual Body Modal Data:

$n_m(k)$ - number of assumed modes for the k^{th} body

$\overline{\mathcal{N}}(k)$ $= n_m(k) + 6$ - number of deformation plus rigid–body degrees of freedom for the k^{th} body

$\eta(k)$ $\in \mathbb{R}^{n_m(k)}$ - vector of modal deformation variables for the k^{th} body

$\Pi_r^j(k)$ $\in \mathbb{R}^6$ - modal spatial displacement vector for the r^{th} mode at the j_k^{th} nodal reference frame

$\Pi^j(k)$ $= \left[\Pi_1^j(k), \cdots, \Pi_{n_m(k)}^j(k)\right] \in \mathbb{R}^{6\times n_m(k)}$ - modal spatial influence vector for the j_k^{th} node. The spatial deformation of node j_k is given by $u(j_k) = \Pi^j(k)\eta(k)$.

$\Pi(k)$ $= \text{col}\left\{\Pi^j(k)\right\} \in \mathbb{R}^{6n_s(k)\times n_m(k)}$ - the modal matrix for the k^{th} body. The r^{th} column of $\Pi(k)$ is denoted $\Pi_r(k) \in \mathbb{R}^{6n_s(k)}$ and is the mode shape function for the r^{th} assumed mode for the k^{th} body. The deformation field for the k^{th} body is given by $u(k) = \Pi(k)\eta(k)$, while $\mathring{u}(k) = \Pi(k)\dot{\eta}(k)$.

$M_m(k)$ $\in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)}$ - modal mass matrix for the k^{th} body.

$K_m(k)$ $\in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)}$ - modal stiffness matrix for the k^{th} body.

Multibody Data:

N - number of bodies in the serial flexible multibody system

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

$n_r(k)$ - number of degrees of freedom for the k^{th} hinge

$\mathcal{N}(k)$ $= n_m(k) + n_r(k)$ - number of deformation plus hinge degrees of freedom for the k^{th} body

$\mathcal{N}$ $= \sum_{k=1}^{N}\mathcal{N}(k)$ - overall deformation plus hinge degrees of freedom for the serial chain

d_k - node on the k^{th} body to which the k^{th} hinge is attached

t_k - node on the k^{th} body to which the $(k-1)^{th}$ hinge is attached

$\mathcal{O}_k$ - reference frame for the k^{th} hinge on the k^{th} body. This frame is fixed to node d_k.

$\mathcal{O}_k^+$ - reference frame for the k^{th} hinge on the $(k+1)^{th}$ body. This frame is fixed to node t_{k+1}.

$\theta(k)$ $\in \mathbb{R}^{n_r(k)}$ - vector of configuration variables for the k^{th} hinge

$\beta(k)$ $\in \mathbb{R}^{n_r(k)}$ - vector of generalized velocities for the k^{th} hinge

$\Delta_V(k)$ $= \begin{pmatrix} \Delta_\omega(k) \\ \Delta_v(k) \end{pmatrix} \in \mathbb{R}^6$ - relative spatial velocity for the k^{th} hinge defined as the spatial velocity of frame $\mathcal{O}_k$ with respect to frame $\mathcal{O}_k^+$

$H^*(k)$ $\in \mathbb{R}^{6\times n_r(k)}$ - joint map matrix for the k^{th} hinge. We have that $\Delta_V(k) = H^*(k)\beta(k)$.

$\vartheta(k)$ $= \begin{pmatrix} \eta(k) \\ \theta(k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}(k)}$ - vector of (deformation plus hinge) generalized configuration variables for the k^{th} body

$\chi(k)$ $= \begin{pmatrix} \dot\eta(k) \\ \beta(k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}(k)}$ - vector of (deformation plus hinge) generalized velocities for the k^{th} body

$V(k)$ $= V(\mathcal{F}_k) = \begin{pmatrix} \omega(k) \\ v(k) \end{pmatrix} \in \mathbb{R}^6$ - spatial velocity of the k^{th} body reference frame $\mathcal{F}_k$, with $\omega(k)$ and $v(k)$ denoting the angular and linear velocities respectively of frame $\mathcal{F}_k$

$V(\mathcal{O}_k)$ $\in \mathbb{R}^6$ - spatial velocity of frame $\mathcal{O}_k$

$V(\mathcal{O}_k^+)$ $\in \mathbb{R}^6$ - spatial velocity of frame $\mathcal{O}_k^+$

$V_s(j_k)$ $\in \mathbb{R}^6$ - spatial velocity of the j^{th} node on the k^{th} body.

$\alpha_s(j_k)$ $\in \mathbb{R}^6$ - spatial acceleration of the j^{th} node on the k^{th} body.

$V_m(k)$ $= \begin{pmatrix} \dot\eta(k) \\ V(k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ - modal spatial velocity of the k^{th} body

$\alpha_m(k)$ $= \dot V_m(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ - modal spatial acceleration of the k^{th} body

$a_m(k)$ $\in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ - modal Coriolis and centrifugal accelerations for the k^{th} body

$b_m(k)$ $\in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ - modal gyroscopic forces for the k^{th} body

$f_m(k)$ $\in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ - modal spatial force of interaction between the k^{th} and $(k+1)^{th}$ bodies

$f_s(j_k)$ $\in \mathbb{R}^6$ - spatial force at node j_k

$f(k)$ $\in \mathbb{R}^6$ - effective spatial force at frame $\mathcal{F}_k$

$T(k)$ $\in \mathbb{R}^{\mathcal{N}(k)}$ - generalized force for the k^{th} body

$H_{\mathcal{F}}(k)$ $= H(k)\phi(\mathcal{O}_k, k) \in \mathbb{R}^{n_r(k)\times 6}$ - joint map matrix referred to frame $\mathcal{F}_k$ for the k^{th} hinge

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

$\mathcal{A}(k)$ $= \begin{pmatrix} [\Pi^t(k)]^* \\ \phi(k, t_k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times 6}$ - relates spatial forces and velocities between node t_k and frame $\mathcal{F}_k$

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

$\Phi(k+1,k)$ $= \mathcal{A}(k+1)\mathcal{B}(k+1,k) = \begin{pmatrix} 0 & [\Pi^t(k+1)]^*\phi(t_{k+1}, k) \\ 0 & \phi(k+1, k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k+1)\times\overline{\mathcal{N}}(k)}$ - the interbody transformation operator which relates modal spatial forces and velocities between the k^{th} and $(k+1)^{th}$ bodies

$$C(k, k-1) = \begin{pmatrix} 0 \\ \vdots \\ \phi(t_k, k-1) \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{6n_s(k)\times 6}$$

$B(k)$ $= [\phi(k, 1_k), \phi(k, 2_k), \cdots, \phi(k, n_s(k))] \in \mathbb{R}^{6\times 6n_s(k)}$ - relates the spatial velocity of frame $\mathcal{F}_k$ to the spatial velocities of all the nodes on the k^{th} body when the body is regarded as being rigid

$\mathcal{M}$ $\in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ - the multibody system mass matrix

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

2 Introduction

This paper uses spatial operators (Refs. 3, 4) to formulate the 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 (see Refs. 5, 6, 7).

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 Section 3 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 Section 4

In Section 5, 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 (Refs. 8, 9) which 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 Section 6 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 Section 7 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 (Ref. 10).

In Section 8 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 Section 9 we discuss the extensions of the formulation and algorithms in this paper to tree and closed-chain topology multibody systems

3 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 k^{th} body is denoted $n_s(k)$. The j^{th} node on the k^{th} body is referred to as the j_k^{th} node. Each body has associated with it a *body reference frame*, denoted $\mathcal{F}_k$ for the k^{th} body. The deformations of the nodes on the body are 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 denoted $u(j_k) \in \mathbb{R}^6$. The overall deformation field for the k^{th} body is defined as the vector $u(k) = \text{col}\left\{u(j_k)\right\} \in \mathbb{R}^{6n_s(k)}$. The vector from frame $\mathcal{F}_k$ to the reference frame on node j_k is denoted $l(k, j_k) \in \mathbb{R}^3$.

With $M_s(j_k) \in \mathbb{R}^{6\times6}$ denoting the spatial inertia of the j^{th} node, the *structural mass matrix* for the k^{th} body $M_s(k)$ is the block diagonal matrix $\text{diag}\left\{M_s(j_k)\right\} \in \mathbb{R}^{6n_s(k)\times 6n_s(k)}$. The *structural stiffness matrix* is denoted $K_s(k) \in \mathbb{R}^{6n_s(k)\times 6n_s(k)}$. Both $M_s(k)$ and $K_s(k)$ are typically generated using finite element analysis.

As shown in Figure 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 towards (away from) the base body. The k^{th} body is attached on the inboard side to the $(k+1)^{th}$ body via the k^{th} hinge, and on the outboard side to the $(k-1)^{th}$ body via 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 referred to as node t_k, while the node to which the inboard hinge (the k^{th} hinge) is attached is denoted node d_k. Thus the k^{th} hinge couples together nodes d_k and t_{k+1}. Attached to each of these pair of adjoining nodes are the k^{th} hinge reference frames denoted $\mathcal{O}_k$ and $\mathcal{O}_k^+$, respectively. The number of degrees of freedom for the k^{th} hinge is denoted $n_r(k)$. The vector of configuration variables for the k^{th} hinge is denoted $\theta(k) \in \mathbb{R}^{n_r(k)}$, while its vector of generalized speeds is denoted $\beta(k) \in \mathbb{R}^{n_r(k)}$. 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 quasi-coordinates 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 k^{th} hinge.

Assumed modes are typically 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 k^{th} 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 for representing 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 (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. 11.

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

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

The r^{th} column of $\Pi(k)$ is denoted $\Pi_r(k)$ and defines the mode shape for the r^{th} assumed mode for the k^{th} 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 $\eta(k) \in \mathbb{R}^{n_m(k)}$ denoting 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 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 k^{th} body are defined as

$$\vartheta(k) \stackrel{\triangle}{=} \begin{pmatrix} \eta(k) \\ \theta(k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}(k)} \quad \text{and} \quad \chi(k) \stackrel{\triangle}{=} \begin{pmatrix} \dot{\eta}(k) \\ \beta(k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}(k)} \tag{3}$$

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

$$\vartheta \stackrel{\triangle}{=} \text{col}\left\{\vartheta(k)\right\} \in \mathbb{R}^{\mathcal{N}} \quad \text{and} \quad \chi \stackrel{\triangle}{=} \text{col}\left\{\chi(k)\right\} \in \mathbb{R}^{\mathcal{N}} \tag{4}$$

where $\mathcal{N} \stackrel{\triangle}{=} \sum_{k=1}^{N} \mathcal{N}(k)$ 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* $T \in \mathbb{R}^{\mathcal{N}}$ 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.

3.1 Recursive Propagation of Velocities

Let $V(k) \in \mathbb{R}^6$ denote the spatial velocity of the k^{th} body reference frame $\mathcal{F}_k$. The spatial velocity $V_s(t_{k+1}) \in \mathbb{R}^6$ of node t_{k+1} (on the inboard 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 $\dot{\eta}(k+1)$ as follows:

$$\begin{aligned} V_s(t_{k+1}) &= \phi^*(k+1, t_{k+1})V(k+1) + \mathring{u}(t_{k+1}) \\ &= \phi^*(k+1, t_{k+1})V(k+1) + \Pi^t(k+1)\dot{\eta}(k+1) \end{aligned} \tag{5}$$

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

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

where $l(x,y) \in \mathbb{R}^3$ denotes the vector between the points x and y. Note that the following important (group) property holds:

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

for arbitrary points x, y and z. As in (5), 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 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 $V(\mathcal{O}_k^+)$ of frame $\mathcal{O}_k^+$ (on the inboard side of the k^{th} hinge) is related to $V_s(t_{k+1})$ via

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

Since the relative spatial velocity $\Delta_V(k)$ across the k^{th} 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 k^{th} hinge is

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

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

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

Putting together (5), (7), (8) and (9), it follows that

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

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

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

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

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

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

where

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

Note that

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

where

$$\mathcal{A}(k) \triangleq \begin{pmatrix} [\Pi^t(k)]^* \\ \phi(k,t_k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times 6} \quad \text{and} \quad \mathcal{B}(k+1,k) \triangleq [0, \quad \phi(t_{k+1},k)] \in \mathbb{R}^{6\times\overline{\mathcal{N}}(k)} \tag{15}$$

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

$$\mathcal{H}(k) = \begin{pmatrix} \mathcal{H}_f(k) \\ \mathcal{H}_r(k) \end{pmatrix} \in \mathbb{R}^{\mathcal{N}(k)\times\overline{\mathcal{N}}(k)} \tag{16}$$

where

$$\mathcal{H}_f(k) \triangleq \Big[I, \quad -[\Pi^d_{\mathcal{F}}(k)]^*\Big] \in \mathbb{R}^{n_m(k)\times\overline{\mathcal{N}}(k)} \quad \text{and} \quad \mathcal{H}_r(k) \triangleq [0, \quad H_{\mathcal{F}}(k)] \in \mathbb{R}^{n_r(k)\times\overline{\mathcal{N}}(k)} \tag{17}$$

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

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

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

$$\Phi \quad \stackrel{\triangle}{=} \quad [I - \mathcal{E}_\Phi]^{-1} = I + \mathcal{E}_\Phi + \cdots + \mathcal{E}_\Phi^{N-1} = \begin{pmatrix} I & 0 & \dots & 0 \\ \Phi(2,1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi(N,1) & \Phi(N,2) & \dots & I \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}} \tag{19}$$

where

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

Also define the spatial operator $\mathcal{H} \stackrel{\triangle}{=} \text{diag}\left\{\mathcal{H}(k)\right\} \in \mathbb{R}^{\mathcal{N} \times \overline{\mathcal{N}}}$. Using these spatial operators, and defining $V_m \stackrel{\triangle}{=} \text{col}\left\{V_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}}$, from (11) it follows that the spatial operator expression for V_m is given by

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

3.2 Modal Mass Matrix for a Single Body

With $V_s(j_k) \in \mathbb{R}^6$ denoting the spatial velocity of node j_k, and $V_s(k) \stackrel{\triangle}{=} \text{col}\left\{V_s(j_k)\right\} \in \mathbb{R}^{6n_s(k)}$ the vector of all nodal spatial velocities for the k^{th} body, it follows (see (5)) that

$$V_s(k) = B^*(k)V(k) + \mathring{u}(k) = [\Pi(k), \quad B^*(k)]V_m(k) \tag{21}$$

where

$$B(k) \stackrel{\triangle}{=} [\phi(k, 1_k), \phi(k, 2_k), \cdots, \phi(k, n_s(k))] \in \mathbb{R}^{6 \times 6n_s(k)} \tag{22}$$

Since $M_s(k)$ is the *structural mass matrix* of the k^{th} body, and usinf (21), the kinetic energy of the k^{th} body can be written in the form

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

where

$$\begin{aligned} M_m(k) \quad &\stackrel{\triangle}{=} \quad \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} M_s(k)[\Pi(k), \quad B^*(k)] = \begin{pmatrix} \Pi^*(k)M_s(k)\Pi(k) & \Pi^*(k)M_s(k)B^*(k) \\ B(k)M_s(k)\Pi(k) & B(k)M_s(k)B^*(k) \end{pmatrix} \\ &= \quad \begin{pmatrix} M_m^{ff}(k) & M_m^{fr}(k) \\ M_m^{rf}(k) & M_m^{rr}(k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k) \times \overline{\mathcal{N}}(k)} \end{aligned} \tag{23}$$

Corresponding to the generalized speeds vector $\chi(k)$, $M_m(k)$ as defined above is the *modal mass matrix* of the k^{th} body. In the block partitioning in (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 $M_m^{fr}(k)$ the flex/rigid coupling block of $M_m(k)$. We will use this notational convention all through this paper. 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 (23) 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 Ref. 12. 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 diagonal.

3.3 Recursive Propagation of Accelerations

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

$$\alpha_m(k) \stackrel{\triangle}{=} \dot{V}_m(k) = \begin{pmatrix} \ddot{\eta}(k) \\ \alpha(k) \end{pmatrix} = \Phi^*(k+1,k)\alpha_m(k+1) + \mathcal{H}^*(k)\dot{\chi}(k) + a_m(k) \tag{24}$$

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

$$a_m(k) = \frac{\boldsymbol{d}\Phi^*(k+1,k)}{\boldsymbol{d}t} V_m(k+1) + \frac{\boldsymbol{d}\mathcal{H}^*(k)}{\boldsymbol{d}t}\chi(k) \tag{25}$$

The detailed expressions for $a_m(k)$ can be found in Ref. 12. Defining $a_m = \text{col}\left\{a_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}}$ and $\alpha_m = \text{col}\left\{\alpha_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}}$, and using spatial operators we can rexpress (24) in the form

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

The vector of spatial accelerations of all the nodes for the k^{th} body, $\alpha_s(k) \stackrel{\triangle}{=} \text{col}\left\{\alpha_s(j_k)\right\} \in \mathbb{R}^{6n_s(k)}$, is obtained by differentiating (21):

$$\alpha_s(k) = \dot{V}_s(k) = [\Pi(k), \quad B^*(k)]\alpha_m(k) + a(k) \tag{27}$$

where

$$a(k) \stackrel{\triangle}{=} \text{col}\left\{a(j_k)\right\} = \frac{\boldsymbol{d}[\Pi(k), \quad B^*(k)]}{\boldsymbol{d}t} V_m(k) \in \mathbb{R}^{6n_s(k)} \tag{28}$$

3.4 Recursive Propagation of Forces

Let $f(k-1) \in \mathbb{R}^6$ 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 $f_s(j_k) \in \mathbb{R}^6$ denoting the spatial force at a node j_k, the force balance equation for node t_k is given by

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

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

$$f_s(j_k) = M_s(j_k)\alpha_s(j_k) + b(j_k) + f_K(j_k) \tag{30}$$

In (29) and (30), $f_K(j_k)$ are the componebts of the vector $f_K(k) = K_s(k)u(k) \in \mathbb{R}^{6n_s(k)}$ denotes the vector of spatial elastic strain forces for the nodes on the k^{th} body, while $b(j_k) \in \mathbb{R}^6$ denotes the spatial gyroscopic force for node j_k and is given by

$$b(j_k) = \begin{pmatrix} \tilde{\omega}(j_k)\mathcal{J}(j_k)\omega(j_k) \\ m(j_k)\tilde{\omega}(j_k)\tilde{\omega}(j_k)p(j_k) \end{pmatrix} \in \mathbb{R}^6 \tag{31}$$

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

$$C(k, k-1) \stackrel{\triangle}{=} \begin{pmatrix} 0 \\ \vdots \\ \phi(t_k, k-1) \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{6n_s(k)\times 6} \quad \text{and} \quad b(k) \stackrel{\triangle}{=} \text{col}\Big\{b(j_k)\Big\} \in \mathbb{R}^{6n_s(k)} \tag{32}$$

it follows from (29) and (30) that

$$f_s(k) \quad = \quad C(k, k-1)f(k-1) + M_s(k)\alpha_s(k) + b(k) + K_s(k)u(k) \tag{33}$$

where $f_s(k) \stackrel{\triangle}{=} \text{col}\Big\{f_s(j_k)\Big\} \in \mathbb{R}^{6n_s(k)}$. Noting that

$$f(k) = B(k)f_s(k) \tag{34}$$

and using the principle of virtual work, it follows from (21) that the *modal spatial forces* $f_m(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)}$ for the k^{th} body are given by

$$f_m(k) \quad \stackrel{\triangle}{=} \quad \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} f_s(k) = \begin{pmatrix} \Pi^*(k)f_s(k) \\ f(k) \end{pmatrix} \tag{35}$$

Premultiplying (33) by $\begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix}$ and using (23), (27), and (35) leads to the following recursive relationship for the modal spatial forces:

$$\begin{aligned} f_m(k) \quad &= \quad \begin{pmatrix} \Pi^*(k)C(k, k-1) \\ B(k)C(k, k-1) \end{pmatrix} f(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \\ &= \quad \begin{pmatrix} [\Pi^t(k)]^* \\ \phi(k, t_k) \end{pmatrix} \phi(t_k, k-1)f(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \\ &= \quad \Phi(k, k-1)f_m(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \end{aligned} \tag{36}$$

Here we have defined

$$b_m(k) \stackrel{\triangle}{=} \begin{pmatrix} \Pi^*(k) \\ B(k) \end{pmatrix} [b(k) + M_s(k)a(k)] \in \mathbb{R}^{\overline{\mathcal{N}}(k)} \tag{37}$$

and the *modal stiffness matrix*

$$K_m(k) \stackrel{\triangle}{=} \begin{pmatrix} \Pi^*(k)K_s(k)\Pi(k) & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k) \times \overline{\mathcal{N}}(k)} \tag{38}$$

The expression for $K_m(k)$ in (38) uses the fact that the columns of $B^*(k)$ are indeed the *deformation dependent* rigid body modes for the k^{th} 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 that

$$K_s(k)B^*(k) = 0 \tag{39}$$

However the common practice (also followed here) of using a constant, deformation–independent structural stiffness matrix leads to the anomalous situation wherein (39) does not hold exactly. We ignore these fictitious extra terms on the left–hand side of (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. 12. From (36), the operator expression for the modal spatial forces $f_m \stackrel{\triangle}{=} \text{col}\left\{f_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}}$ for all the bodies in the chain is given by

$$f_m = \Phi(M_m\alpha_m + b_m + K_m\vartheta) \tag{40}$$

where

$$M_m \stackrel{\triangle}{=} \text{diag}\left\{M_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}},\ K_m \stackrel{\triangle}{=} \text{diag}\left\{K_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}} \times \overline{\mathcal{N}}},\ \text{and } b_m \stackrel{\triangle}{=} \text{col}\left\{b_m(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}}$$

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

$$T = \mathcal{H}f_m \tag{41}$$

3.5 Operator Expression for the System Mass Matrix

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

$$\begin{aligned} V_m &= \Phi^*\mathcal{H}^*\chi \\ \alpha_m &= \Phi^*(\mathcal{H}^*\dot{\chi} + a_m) \\ f_m &= \Phi(M_m\alpha_m + b_m + K_m\vartheta) = \Phi M_m\Phi^*\mathcal{H}^*\dot{\chi} + \Phi(M_m\Phi^*a_m + b_m + K_m\vartheta) \\ T &= \mathcal{H}f_m = \mathcal{H}\Phi M_m\Phi^*\mathcal{H}^*\dot{\chi} + \mathcal{H}\Phi(M_m\Phi^*a_m + b_m) \\ &= \mathcal{M}\dot{\chi} + \mathcal{C} \end{aligned} \tag{42}$$

where

$$\mathcal{M} \stackrel{\triangle}{=} \mathcal{H}\Phi M_m \Phi^* \mathcal{H}^* \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \quad \text{and} \quad \mathcal{C} \stackrel{\triangle}{=} \mathcal{H}\Phi(M_m\Phi^* a_m + b_m + K_m\vartheta) \in \mathbb{R}^{\mathcal{N}} \tag{43}$$

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

It is noteworthy that the operator expressions for $\mathcal{M}$ and $\mathcal{C}$ are identical in form to those for rigid multibody systems (see Refs. 3, 13). 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. This is the approach adopted here.

4 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 upon composite body inertias or the conjugate gradient method (Ref. 14).

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

$$\left\{\begin{array}{rcl} & & V_m(N+1) = 0, \quad \alpha_m(N+1) = 0 \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{N \cdots 1} \\ V_m(k) & = & \Phi^*(k+1,k)V_m(k+1) + \mathcal{H}^*(k)\chi(k) \\ \alpha_m(k) & = & \Phi^*(k+1,k)\alpha_m(k+1) + \mathcal{H}^*(k)\dot{\chi}(k) + a_m(k) \\ \textbf{end loop} & & \end{array}\right.$$

$$\left\{\begin{array}{rcl} & & f_m(0) = 0 \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{1 \cdots N} \\ f_m(k) & = & \Phi(k,k-1)f_m(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \\ T(k) & = & \mathcal{H}(k)f_m(k) \\ \textbf{end loop} & & \end{array}\right. \tag{44}$$

The structure of this algorithm closely resembles the recursive Newton-Euler inverse dynamics algorithm for rigid multibody systems (see Refs. 15, 3). All external forces on the k^{th} body are handled by absorbing them into the gyroscopic force term $b_m(k)$. Base mobility is handled by attaching an additional 6 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 $\mathcal{H}(k)$ in (12) and (13), the Newton–Euler recursions in (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 that

$$V_m(k) = \begin{pmatrix} V_m^f(k) \\ V_m^r(k) \end{pmatrix}, \ \alpha_m(k) = \begin{pmatrix} \alpha_m^f(k) \\ \alpha_m^r(k) \end{pmatrix}, \ f_m(k) = \begin{pmatrix} f_m^f(k) \\ f_m^r(k) \end{pmatrix}, \text{ and } T(k) = \begin{pmatrix} T^f(k) \\ T^r(k) \end{pmatrix}$$

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

$$\begin{cases} \qquad V_m(N+1) = 0, \quad \alpha_m(N+1) = 0 \\ \textbf{for } \boldsymbol{k} \ = \ \boldsymbol{N} \cdots \mathbf{1} \\ \quad V_m^f(k) \ = \ \dot{\eta}(k) \\ \quad V_m^r(k) \ = \ \phi^*(t_{k+1}, k)\mathcal{A}^*(k+1)V_m(k+1) + H_{\mathcal{F}}^*(k)\beta(k) - \Pi_{\mathcal{F}}^d(k)\dot{\eta}(k) \\ \\ \quad \alpha_m^f(k) \ = \ \ddot{\eta}(k) \\ \quad \alpha_m^r(k) \ = \ \phi^*(t_{k+1}, k)\mathcal{A}^*(k+1)\alpha_m(k+1) + H_{\mathcal{F}}^*(k)\dot{\beta}(k) - \Pi_{\mathcal{F}}^d(k)\ddot{\eta}(k) + a_m^r(k) \\ \textbf{end loop} \end{cases} \tag{45}$$

$$\begin{cases} \qquad f_m(0) = 0 \\ \textbf{for } \boldsymbol{k} \ = \ \mathbf{1} \cdots \boldsymbol{N} \\ \quad f_m(k) \ = \ \mathcal{A}(k)\phi(t_k, k-1)f_m^r(k-1) + M_m(k)\alpha_m(k) + b_m(k) + K_m(k)\vartheta(k) \\ \quad T(k) \ = \ \begin{pmatrix} T^f(k) \\ T^r(k) \end{pmatrix} = \begin{pmatrix} f_m^f(k) - [\Pi_{\mathcal{F}}^d(k)]^* f_m^r(k) \\ H_{\mathcal{F}}(k) f_m^r(k) \end{pmatrix} \\ \textbf{end loop} \end{cases}$$

Flexible multibody systems have actuators typically only at the hinges. Thus for the k^{th} body, only the subset of the generalized forces vector $T(k)$ corresponding to the hinge actuator forces $T^r(k)$ can be set, while the remaining generalized forces $T^f(k)$ are zero. Thus in contrast with rigid multibody systems, flexible multibody systems are *under-actuated systems* (Ref. 16), since the number of available actuators is less than the number of motion degrees of freedom in the system. For such under-actuated 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 $T^f(.) = 0$.

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

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

Later in Section 6 we describe the recursive articulated body forward dynamics algorithm that does not require the explicit computation of either $\mathcal{M}$ or $\mathcal{C}$

It is evident from (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 (45). We describe next an efficient composite–body–based recursive algorithm for the computation of the mass matrix $\mathcal{M}$. This algorithm is based upon the following lemma which contains a decomposition of the mass matrix into block diagonal, block upper triangular and block lower triangular components.

Lemma 5.1 *Define the composite body inertias* $R(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)}$ *recursively for all the bodies in the serial chain as follows:*

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

Also define $R \stackrel{\triangle}{=} \operatorname{diag}\left\{R(k)\right\} \in \mathbb{R}^{\overline{\mathcal{N}}\times\overline{\mathcal{N}}}$. *Then we have the following spatial operator decomposition*

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

where $\tilde{\Phi} \stackrel{\triangle}{=} \Phi - I$.

Proof: See Appendix A. ∎

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 (43) and Lemma 5.1 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 (49) are block diagonal, block lower triangular and block upper triangular respectively. The following algorithm for computing the mass matrix $\mathcal{M}$ computes the elements of these terms recursively:

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

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. The structure of this algorithm closely resembles the composite rigid body algorithm for computing the mass matrix of rigid multibody systems (Refs. 14, 10). 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.

6 Factorization and Inversion of the Mass Matrix

An operator factorization of the system mass matrix $\mathcal{M}$, denoted the *Innovations Operator Factorization*, is derived in this section. This factorization is an alternative to the Newton–Euler factorization in (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. 3).

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

$$\begin{cases} \qquad\qquad P^+(0) = 0 \\ \textbf{for } \boldsymbol{k} \ = \ \mathbf{1} \cdots \boldsymbol{N} \\ \qquad P(k) \ = \ \Phi(k,k-1)P^+(k-1)\Phi^*(k,k-1) + M_m(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\ \qquad D(k) \ = \ \mathcal{H}(k)P(k)\mathcal{H}^*(k) \in \mathbb{R}^{\mathcal{N}(k)\times\mathcal{N}(k)} \\ \qquad G(k) \ = \ P(k)\mathcal{H}^*(k)D^{-1}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\mathcal{N}(k)} \\ \qquad K(k+1,k) \ = \ \Phi(k+1,k)G(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\mathcal{N}(k)} \\ \qquad \overline{\tau}(k) \ = \ I - G(k)\mathcal{H}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\ \qquad P^+(k) \ = \ \overline{\tau}(k)P(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\ \qquad \Psi(k+1,k) \ = \ \Phi(k+1,k)\overline{\tau}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\ \textbf{end loop} \end{cases} \tag{51}$$

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

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

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

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

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

where

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

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 lemma.

Lemma 6.1

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

Proof: See Appendix A. ∎

Note that the factor $[I + \mathcal{H}\Phi K] \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ is square, block lower triangular and nonsingular, while D is a block diagonal matrix. This factorization provides a closed–form expression for the 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 6.2

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

Proof: See Appendix A. ∎

It follows from Lemmas 6.1 and 6.2 that the operator expression for the inverse of the mass matrix is given by:

Lemma 6.3

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

∎

Once again, note that the factor $[I - \mathcal{H}\Psi K]$ is square, block lower triangular and nonsingular and so Lemma 6.3 provides a closed–form expression for the block LDL^* decomposition of $\mathcal{M}^{-1}$.

7 Articulated Body Forward Dynamics Algorithm

We first use the operator expression for the mass matrix inverse developed in Section 6 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 7.1

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

Proof: See Appendix A. ∎

As in the case of rigid multibody systems (Refs. 3, 4), the direct recursive implementation of (57) leads to the following recursive forward dynamics algorithm:

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

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

The structure of this algorithm is closely related to the structure of the well known Kalman filtering and smoothing algorithms (Ref. 10). All the degrees of freedom for each body (as characterized by its joint map matrix $\mathcal{H}^*(.)$) 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.

7.1 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 (11)

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

and the partitioned form of $\mathcal{H}(k)$ in (13)

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

Introducing a dummy variable k', we can rewrite (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

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

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 and 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 ones. The new $\mathcal{H}^*$ operator now has the same number of columns but twice the number of rows as the original $\mathcal{H}^*$ operator. The new Φ operator has twice as many rows and columns as the original one. Repeating the analysis described in the previous sections, we once again obtain the same operator expression as (57). This expression also leads to a recursive forward dynamics algorithm as in (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 (51)) for computing the articulated body quantities is as follows:

$$\left\{\begin{array}{rcl}
 & P^+(0) = 0 & \\
\textbf{for } \boldsymbol{k} & \boldsymbol{=} & \boldsymbol{1 \cdots N} \\
\Gamma(k) &=& \mathcal{B}(k,k-1)P^+(k-1)\mathcal{B}^*(k,k-1) \in \mathbb{R}^{6\times 6} \\
P(k) &=& \mathcal{A}(k)\Gamma(k)\mathcal{A}^*(k) + M_m(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
D_f(k) &=& \mathcal{H}_f(k)P(k)\mathcal{H}_f^*(k) \in \mathbb{R}^{n_m(k)\times n_m(k)} \\
G_f(k) &=& P(k)\mathcal{H}_f^*(k)D_f^{-1}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times n_m(k)} \\
\overline{\tau}_f(k) &=& I - G_f(k)\mathcal{H}_f(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
P_r(k) &=& \overline{\tau}_f(k)P(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
 & & \\
D_r(k) &=& \mathcal{H}_r(k)P_r(k)\mathcal{H}_r^*(k) \in \mathbb{R}^{n_r(k)\times n_r(k)} \\
G_r(k) &=& P_r(k)\mathcal{H}_r^*(k)D_r^{-1}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times n_r(k)} \\
\overline{\tau}_r(k) &=& I - G_r(k)\mathcal{H}_r(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
P^+(k) &=& \overline{\tau}_r(k)P_r(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
\Psi(k+1,k) &=& \Phi(k+1,k)\overline{\tau}(k) \in \mathbb{R}^{\overline{\mathcal{N}}(k)\times\overline{\mathcal{N}}(k)} \\
\textbf{end loop} & &
\end{array}\right. \tag{62}$$

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

$$
\begin{aligned}
\Gamma(k) &= \phi(t_k, k-1) P_R^+(k-1) \phi^*(t_k, k-1), \qquad (P_R^+(k) \text{ is defined below}) \\
G_f(k) &= \begin{pmatrix} \times \\ g(k) \end{pmatrix}, \quad \text{where} \quad g(k) = \mu(k) D_f^{-1}(k) \in \mathbb{R}^{6 \times n_m(k)}, \\
& \qquad \text{and} \quad \mu(k) \stackrel{\triangle}{=} [P^{rf}(k), \quad P^{rr}(k)] \mathcal{H}_f^*(k) \in \mathbb{R}^{6 \times n_m(k)} \\
P_r(k) &= \begin{pmatrix} \times & \times \\ \times & P_R(k) \end{pmatrix}, \quad \text{where} \quad P_R(k) = P^{rr}(k) - g(k)\mu^*(k) \in \mathbb{R}^{6 \times 6} \\
D_r(k) &= H_{\mathcal{F}}(k) P_R(k) H_{\mathcal{F}}^*(k) \in \mathbb{R}^{n_r(k) \times n_r(k)} \\
G_r(k) &= \begin{pmatrix} \times \\ G_R(k) \end{pmatrix}, \quad \text{where} \quad G_R(k) \stackrel{\triangle}{=} P_R(k) H_{\mathcal{F}}^*(k) D_r^{-1}(k) \in \mathbb{R}^{6 \times n_r(k)} \\
\overline{\tau}_r(k) &= \begin{pmatrix} I & \times \\ 0 & \overline{\tau}_R(k) \end{pmatrix}, \quad \text{where} \quad \overline{\tau}_R(k) = I - G_R(k) H_{\mathcal{F}}(k) \in \mathbb{R}^{6 \times 6} \\
P^+(k) &= \begin{pmatrix} \times & \times \\ \times & P_R^+(k) \end{pmatrix}, \quad \text{where} \quad P_R^+(k) = \overline{\tau}_R(k) P_R(k) \in \mathbb{R}^{6 \times 6}
\end{aligned}
$$

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

$$
\left\{
\begin{aligned}
& \qquad P_R^+(0) = 0 \\
& \textbf{for } \boldsymbol{k} \ = \ \boldsymbol{1 \cdots N} \\
& \qquad \Gamma(k) = \phi(t_k, k-1) P_R^+(k-1) \phi^*(t_k, k-1) \\
& \qquad P(k) = \mathcal{A}(k) \Gamma(k) \mathcal{A}^*(k) + M_m(k) \\
& \qquad D_f(k) = \mathcal{H}_f(k) P(k) \mathcal{H}_f^*(k) \\
& \qquad \mu(k) = [P^{rf}(k), \quad P^{rr}(k)] \mathcal{H}_f^*(k) \\
& \qquad g(k) = \mu(k) D_f^{-1}(k) \\
& \qquad P_R(k) = P^{rr}(k) - g(k)\mu^*(k) \\
& \qquad D_R(k) = H_{\mathcal{F}}(k) P_R(k) H_{\mathcal{F}}^*(k) \\
& \qquad G_R(k) = P_R(k) H_{\mathcal{F}}^*(k) D_R^{-1}(k) \\
& \qquad \overline{\tau}_R(k) = I - G_R(k) H_{\mathcal{F}}(k) \\
& \qquad P_R^+(k) = \overline{\tau}_R(k) P_R(k) \\
& \textbf{end loop}
\end{aligned}
\right.
\tag{63}
$$

7.2 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 (57). The algorithm consists of the following steps: (a) a base–to–tip recursion as in (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 the bodies; (b) computation of the articulated body quantities using (78) and (63); and (c) a tip-to-base recursion followed by a base-to-tip recursion for the joint accelerations $\dot{\chi}$ as described below:

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

$$\left\{\begin{array}{rcl} & & \alpha_m(N+1) = 0 \\ \textbf{for } \boldsymbol{k} & = & \boldsymbol{N} \cdots \mathbf{1} \\ \alpha_R^+(k) & = & \phi^*(t_{k+1}, k)\mathcal{A}^*(k+1)\alpha_m(k+1) \in \mathbb{R}^6 \\ \dot{\beta}(k) & = & \nu_R(k) - G_R^*(k)\alpha_R^+(k) \in \mathbb{R}^{n_r(k)} \\ \alpha_R(k) & = & \alpha_R^+(k) + H_{\mathcal{F}}^*(k)\dot{\beta}(k) + a_{mR}(k) \in \mathbb{R}^6 \\ \ddot{\eta}(k) & = & \nu_f(k) - g^*(k)\alpha_R(k) \in \mathbb{R}^{n_m(k)} \\ \alpha_m(k) & = & \begin{pmatrix} \ddot{\eta}(k) \\ \alpha_R(k) - \Pi_{\mathcal{F}}^d(k)\ddot{\eta}(k) \end{pmatrix} \in \mathbb{R}^{\overline{\mathcal{N}}(k)} \\ \textbf{end loop} & & \end{array}\right. \tag{64}$$

The recursion in (64) is obtained by simplifying the recursions in (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 described in Section 5, the articulated body forward dynamics algorithm does not require the explicit computation of either $\mathcal{M}$ 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 references (Refs. 17, 3).

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 (Ref. 8). The results from the two independent simulations have shown complete agreement.

8 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. 18 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 (non–extremal) flexible body, denoted the k^{th} 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 non–extremal 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 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.

8.1 Computational Cost of the Composite Body Forward Dynamics Algorithm

The composite body forward dynamics algorithm described in Section 5 is based on solving the linear matrix equation

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

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

1. Cost of computing $R(k)$ for the k^{th} body using the algorithm in (50) is $[48n_m(k) + 90]\mathrm{M} + [n_m^2(k) + \frac{97}{2}n_m(k) + 116]\mathrm{A}$
2. Contribution of the k^{th} body to the cost of computing $\mathcal{M}$ (excluding cost of $R(k)$'s) using the algorithm in (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 $\mathcal{C}$ can be obtained by using the inverse dynamics algorithm described in (45) for computing the generalized forces T. The contribution of the k^{th} body to the computational cost for $\mathcal{C}(k)$ 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 $T - \mathcal{C}$ is $\{\mathcal{N}\}\,\mathrm{A}$.
5. The cost of solving the linear equation in (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 $O(\mathcal{N}^3)$.

8.2 Computational Cost of the Articulated Body Forward Dynamics Algorithm

The articulated body forward dynamics algorithm is based on the recursions described in (78), (63) and (64). Since the computations in (78) 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 (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 (77) 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 (63) for the k^{th} body based on the explicit inversion of $D_f(k)$ (used when $n_m(k) \leq 7$) is $\left\{\frac{5}{6}n_m^3(k) + \frac{25}{2}n_m^2(k) + \frac{764}{3}n_m(k) + 180\right\}\mathrm{M} + \left\{\frac{5}{6}n_m^3(k) + \frac{21}{2}n_m^2(k) + \frac{548}{3}n_m(k) + 164\right\}\mathrm{A}$.
   - Cost of (63) for the k^{th} body based on the indirect computation of $D_f^{-1}(k)$ (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 (64) for the k^{th} body is $\{n_m^2(k) + 25n_m(k) + 49\}\,\mathrm{M} + \{n_m^2(k) + 24n_m(k) + 50\}\,\mathrm{A}$.
3. The cost for the base-to-tip recursion sweep in (64) for the k^{th} body is $\{18n_m(k) + 52\}\,\mathrm{M} + \{19n_m(k) + 42\}\,\mathrm{A}$.

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 versus the number of assumed–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 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.

9 Extensions to General Topology Flexible Multibody Systems

For rigid multibody systems, Ref. 13 describes the extensions to the dynamics formulation and algorithms that are required as the topology 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. 13 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 inwards 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. 13, 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. 13.

10 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. While 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.

11 Acknowledgements

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

References

[3]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, No. 4, Aug. 1991, pp. 371–381.

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

[5]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.

[6]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, No. 3, Sept. 1988, pp. 687–693.

[7]Keat, J.E., "Multibody System Order n Dynamics Formulation Based on Velocity Transform Method," *Journal of Guidance, Control and Dynamics*, Vol. 13, No. 2, March-April 1990.

[8]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.

[9]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, Sept. 1985, pp. 584–590.

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

[11]Likins, P.W., "Modal Method for Analysis of Free Rotations of Spacecraft," *AIAA Journal*, Vol. 5, July 1967, pp. 1304–1308.

[12]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.

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

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

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

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

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

[18] 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).

[19] 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, March–April 1987, pp. 139–151.

Appendix A: Proofs of the Lemmas

At the operator level, the proofs of the lemmas in this publication are completely analogous to those for rigid multibody systems (Refs. 3, 4).

Proof of Lemma 5.1: Using operators, we can rewrite (47) in the form

$$M_m = R - \mathcal{E}_\Phi R \mathcal{E}_\Phi^* \tag{65}$$

From (19) it follows that $\Phi\mathcal{E}_\Phi = \mathcal{E}_\Phi\Phi = \Phi - I = \tilde{\Phi}$. Multiplying (65) from the left and right by Φ and Φ^* respectively leads to

$$\Phi M_m \Phi^* = \Phi R \Phi^* - \Phi\mathcal{E}_\Phi R \mathcal{E}_\Phi^* \Phi^* = (\tilde{\Phi} + I)R(\tilde{\Phi} + I)^* - \tilde{\Phi}R\tilde{\Phi}^* = R + \tilde{\Phi}R + R\tilde{\Phi}^*$$

$\blacksquare$

Proof of Lemma 6.1: It is easy to verify that $\overline{\tau}P\overline{\tau}^* = \overline{\tau}P$. As a consequence, the recursion for $P(.)$ in (51) can be rewritten in the form

$$M_m \quad = \quad P - \mathcal{E}_\Psi P \mathcal{E}_\Psi^* = P - \mathcal{E}_\Psi P \mathcal{E}_\Phi^* = P - \mathcal{E}_\Phi P \mathcal{E}_\Phi^* + KDK^* \tag{66}$$

Pre– and post–multiplying (66) by Φ and Φ^* respectively then leads to

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

Hence,

$$\Longrightarrow \mathcal{M} = \mathcal{H}\Phi M_m \Phi^* \mathcal{H}^* = \mathcal{H}[P + \tilde{\Phi}P + P\tilde{\Phi}^* + \Phi K D K^* \Phi^*]\mathcal{H}^*$$
$$= D + \mathcal{H}\Phi K D + D K^* \Phi^* \mathcal{H}^* + \mathcal{H}\Phi K D K^* \Phi^* \mathcal{H}^* = [I + \mathcal{H}\Phi K] D [I + \mathcal{H}\Phi K]^*$$

∎

Proof of Lemma 6.2: Using a standard matrix identity we have that

$$[I + \mathcal{H}\Phi K]^{-1} = I - \mathcal{H}\Phi[I + K\mathcal{H}\Phi]^{-1} K \tag{67}$$

Note that

$$\Psi^{-1} = I - \mathcal{E}_\Psi = (I - \mathcal{E}_\Phi) + \mathcal{E}_\Phi G \mathcal{H} = \Phi^{-1} + K\mathcal{H} \tag{68}$$

from which it follows that

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

Using this with (67) it follows that

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

∎

Proof of Lemma 7.1: From (42) and (43), the expression for the generalized accelerations $\dot{\chi}$ is given by

$$\dot{\chi} = \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}) = [I - \mathcal{H}\Psi K]^* D^{-1} [I - \mathcal{H}\Psi K]\Big[\mathcal{T} - \mathcal{H}\Phi[M_m \Phi^* a_m + b_m + K_m \vartheta]\Big] \tag{69}$$

From (68) we have that

$$[I - \mathcal{H}\Psi K]\mathcal{H}\Phi = \mathcal{H}\Psi[\Psi^{-1} - K\mathcal{H}]\Phi = \mathcal{H}\Psi \tag{70}$$

Thus (69) can be written as

$$\dot{\chi} = [I - \mathcal{H}\Psi K]^* D^{-1}\Big[\mathcal{T} - \mathcal{H}\Psi[K\mathcal{T} + M_m \Phi^* a_m + b_m + K_m \vartheta]\Big] \tag{71}$$

From (66) it follows that

$$M_m = P - \mathcal{E}_\Psi P \mathcal{E}_\Phi^* \implies \Psi M_m \Phi^* = \Psi P + P\tilde{\Phi}^* \tag{72}$$

and so (71) simplifies to

$$\dot{\chi} = [I - \mathcal{H}\Psi K]^* D^{-1}\Big[\mathcal{T} - \mathcal{H}\Psi[K\mathcal{T} + P a_m + b_m + K_m \vartheta] - \mathcal{H}P\tilde{\Phi}^* a_m\Big] \tag{73}$$

From (68) we have that

$$[I - \mathcal{H}\Psi K]^* D^{-1} \mathcal{H} P \tilde{\Phi}^* = [I - \mathcal{H}\Psi K]^* K^* \Phi^* = K^* \Psi^* [\Psi^{-*} - K\mathcal{H}]^* \Phi^* = K^* \Psi^* \tag{74}$$

Using (74) in (73) leads to the result. ∎

Appendix B: Ruthless Linearization of Flexible Body Dynamics

It has been pointed out in recent literature (Refs. 19, 18) 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 (Ref. 18). 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:

$$M_m(k) \approx M_m^0(k), \quad a_m(k) \approx \begin{pmatrix} 0 \\ a_{mR}^0(k) \end{pmatrix}, \quad \text{and} \quad b_m(k) \approx b_m^0(k) \tag{75}$$

With this approximation, $M_m(k)$ is constant in the body frame, while $a_m(k)$ 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 (63) can be simplified. Using the matrix identity

$$[A + BCB^*]^{-1} = A^{-1} - A^{-1}B[C^{-1} + B^*A^{-1}B]^{-1}B^*A^{-1} \tag{76}$$

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

$$D_f^{-1}(k) = \Lambda(k) - \Upsilon(k)[\Gamma^{-1}(k) + \Omega(k)]^{-1}(k)\Upsilon^*(k) \tag{77}$$

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

$$\left\{ \begin{array}{rcl} \textbf{for } \boldsymbol{k} & \boldsymbol{=} & \boldsymbol{1 \cdots N} \\ \Lambda(k) & = & [\mathcal{H}_f(k)M_m(k)\mathcal{H}_f^*(k)]^{-1} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}} \\ \zeta(k) & = & \mathcal{H}_f(k)\mathcal{A}(k) \in \mathbb{R}^{\mathcal{N}\times 6} \\ \Upsilon(k) & = & \Lambda(k)\zeta(k) \in \mathbb{R}^{\mathcal{N}\times 6} \\ \Omega(k) & = & \zeta^*(k)\Upsilon(k) \in \mathbb{R}^{6\times 6} \\ \textbf{end loop} & & \end{array} \right. \tag{78}$$

Using (77) reduces the computational cost for computing the articulated body inertias to a quadratic rather than a cubic function of the number of modes.

Figure Captions

1. Illustration of links and hinges in a flexible serial multibody system
2. A comparison of the computational cost in floating point operations for the articulated body and the composite body forward dynamics algorithms for a serial chain multibody systems with 10 flexible bodies.

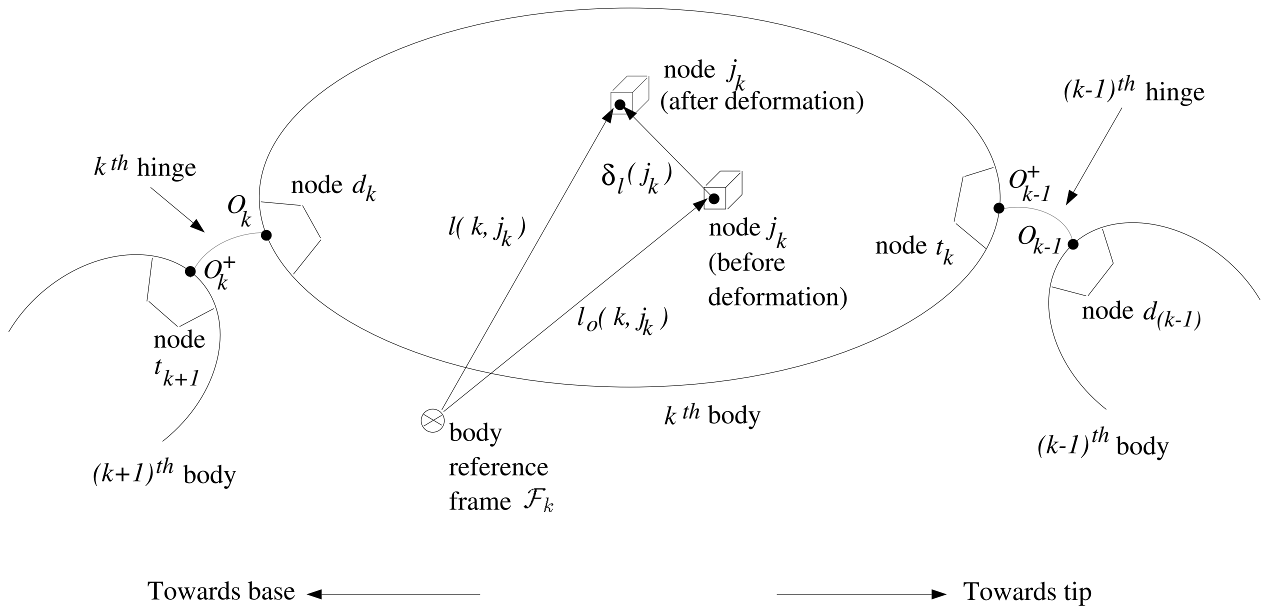

Figure 1: Illustration of links and hinges in a flexible serial multibody system

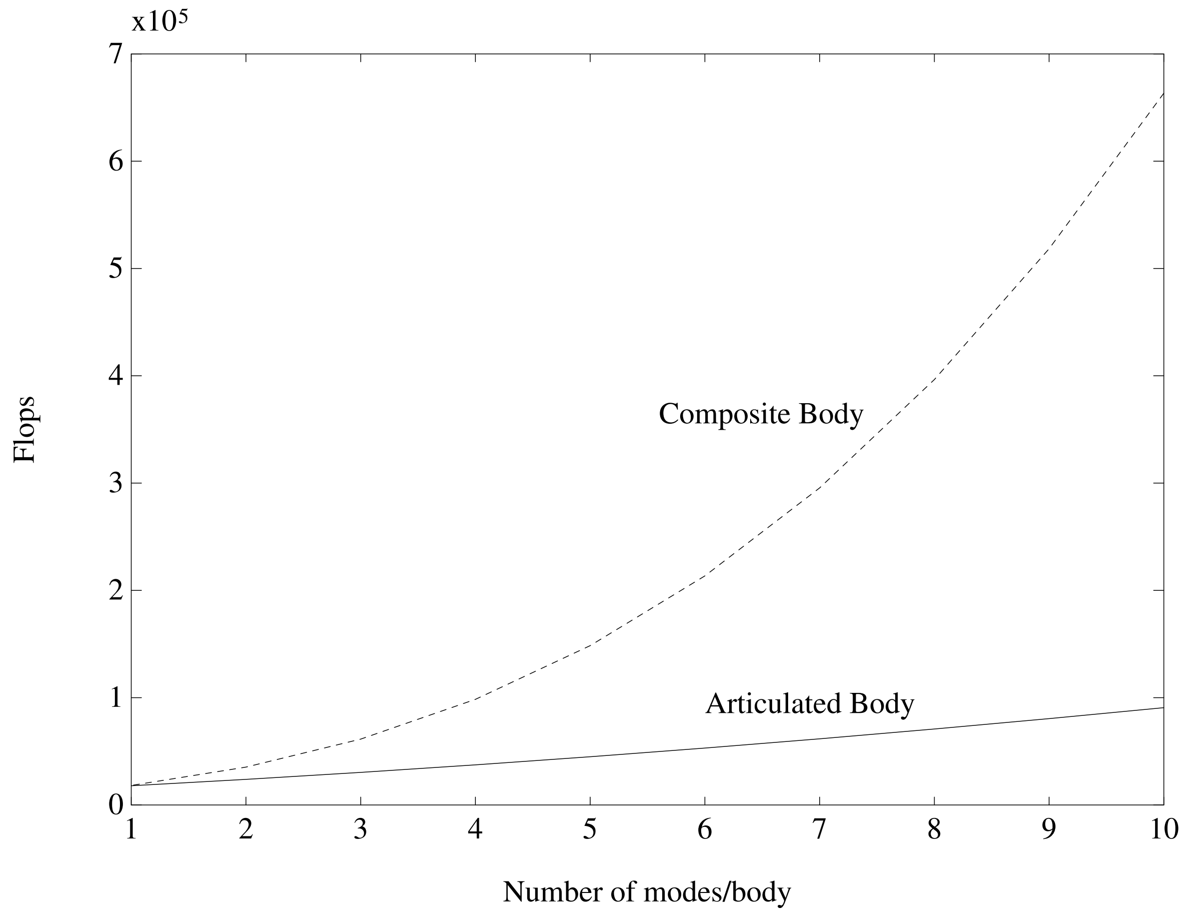

Figure 2: A comparison of the computational cost in floating point operations for the articulated body and the composite body forward dynamics algorithms for a serial chain multibody systems with 10 flexible bodies.