The Journal of the Astronautical Sciences, Vol. 40, No. 1, January-March 1992, pp. 27–50

Spatial Operator Algebra for Multibody System Dynamics[1]

G. Rodriguez,[2] A. Jain,[2] and K. Kreutz-Delgado[3]

Abstract

This paper describes a new *spatial operator algebra* for the dynamics of general-topology rigid multibody systems. Spatial operators allow a concise and systematic formulation of the dynamical equations of motion of multibody systems and the development of efficient computational algorithms. Equations of motion are developed for progressively more complex systems: serial chains, topological trees, and closed-loop systems. New operator factorizations and expressions for the mass matrix and its inverse are derived and used to obtain efficient, spatially recursive computational algorithms. The algorithms can be easily reconfigured in response to changes in the constraints and the topology of constituent bodies. Thus, they are particularly suited for time-varying multibody systems. References are provided for extensions to flexible multibody systems. Spatially recursive algorithms, based on the sequential filtering and smoothing methods encountered in optimal estimation theory, provide the computational infrastructure to mechanize the spatial operators.

Introduction

There are two major current challenges in multibody system dynamics: achieving computational efficiency for increasingly complex systems; and retaining algorithmic efficiency while accounting for possible event-dependent changes to the constraints and topology of the constituent bodies. These challenges occur for example in highly complex and interactive spacecraft and in space robotic systems which require that the dynamics algorithms be quickly reconfigured in response to configuration changes without sacrificing computational efficiency.

In recent years, there has been significant progress in the fomulation of the equations of motion and on the development of efficient dynamics algorithms for multibody systems. A large part of this work, [1]–[6], has focused on the develop-

[1]Presented at the 3rd Annual Conference on Aerospace Computational Control, August 28–30, 1989, Oxnard, California.

[2]Jet Propulsion Laboratory/California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109.

[3]AMES Department, University of California, San Diego, La Jolla, CA 92093.

ment of recursive computational algorithms for the dynamics of serial rigid multibody systems. References [5, 7] describe recursive forward dynamics algorithms for tree topology rigid multibody systems. References [8, 9] contain recursive algorithms specific to closed topology rigid multibody systems consisting of multiple robot manipulators grasping a rigid object. References [10, 11] describe recursive algorithms for general topology rigid multibody systems.

The *spatial operator algebra* [8, 9] arose from a recognition of the close parallels between the structure of the equations of motion for serial chain dynamics and those encountered in the area of optimal estimation theory [12]. These parallels naturally led to the introduction of spatial operators to obtain a concise formulation of the equations of motion of multibody systems. The spatial operators have been used to obtain new operator factorizations and expressions for the mass matrix and its inverse. These operator expressions make it possible to recognize high-level mathematical patterns associated with the mass matrix which the detailed algorithms do not reveal. Therefore, the number of symbols that the analyst has to see is reduced significantly. In addition, the spatial operators are mechanized by very efficient spatially recursive computational algorithms whose complexity depends only linearly on the number of degrees of freedom (dofs). These algorithms closely resemble the algorithms used for recursive filtering and smoothing in the Kalman filtering and estimation problems. There is the reassuring presence of such familiar concepts as Riccati equations and Kalman gains. This makes it easier to mechanize the dynamics algorithms and to monitor their numerical stability and robustness. Much of the experience gained over many years of research in estimation theory can now be used to solve multibody system dynamics problems.

In this paper, the spatial operator algebra is advanced as a new systematic procedure for concisely formulating the equations of motion and deriving efficient spatially recursive dynamics algorithms for general topology rigid multibody systems. Spatial operator formulations of the dynamics of serial chain, tree and closed chain topology multibody systems are developed in a progressive sequence. It is seen that the spatial operator formulation of the dynamics for serial and tree topology systems are identical in form. Consequently, such results as the operator factorization and inversion of the mass matrix for serial chains are directly applicable to tree topology systems. Moreover, the recursive computational algorithms for serial chains extend with only minor modifications to tree topology systems.

Closed topology systems are modeled as consisting of *primary* and *secondary* tree-topology subsystems along with additional kinematical closure constraints within and between them. The primary system represents the major part of the system whose topology and constituent bodies do not change with time. The secondary system is much smaller in most applications, and represents the part of the system whose topology and constituent bodies do change with time.

A recursive computational algorithm is then developed for closed-chain systems. This algorithm contains separate steps for the dynamics of the primary and secondary systems using recursive algorithms for tree-topology systems, plus additional steps to handle the closure constraints. A significant feature of these

algorithms is that only relatively small and localized changes are needed for reconfiguration in response to changes in either the topology, the constraints or the constituent bodies of the system. The overall computational complexity of the algorithm stays linear in the number of degrees of freedom in the system. In addition, there is a linear dependency (in the absence of kinematical singularities) on the number of closed loops in the system.

Extensions of the spatial operator algebra and the computational algorithms to flexible multibody system dynamics are discussed in [13].

6-Dimensional Spatial Notation

Coordinate-free 6-dimensional *spatial notation* is used throughout this paper. Given the linear and angular velocities v and ω, the linear force F, and moment N in $\mathcal{R}^3$ at a point on a body, the (inertial) *spatial velocity* V, (inertial) *spatial acceleration* α and the *spatial force* f in $\mathcal{R}^6$ are:

$$V \triangleq \begin{pmatrix} \omega \\ v \end{pmatrix}, \qquad \alpha \triangleq \dot{V}, \qquad f \triangleq \begin{pmatrix} N \\ F \end{pmatrix}$$

Here " $\dot{}$ " implies the time derivative in an inertial frame. The *rigid body transformation operator* $\phi(x, y) \in \mathcal{R}^{6\times 6}$ for two points x and y is:

$$\phi(l_{x,y}) = \phi(x, y) \triangleq \begin{pmatrix} I & \tilde{l}_{x,y} \\ 0 & I \end{pmatrix}$$

where $l_{x,y} \in \mathcal{R}^3$ is the vector joining the two points x and y. $\tilde{l}_{x,y}$ is the cross-product matrix associated with $l_{x,y}$ which acts on a vector to produce the cross-product of $l_{x,y}$ with the vector. $\phi(x, y)$ and $\phi^*(x, y)$ transform spatial forces and spatial velocities respectively between the two body-fixed points x and y on a rigid body. The spatial inertia $M_O \in \mathcal{R}^{6\times 6}$ of a rigid body at a point O on the body is

$$M(O) \triangleq \begin{pmatrix} \mathcal{I}(O) & m\tilde{p} \\ -m\tilde{p} & mI \end{pmatrix}$$

where $p \in \mathcal{R}^3$ is the vector from O to the center of mass of the body, m is the mass of the body, and $\mathcal{I}(O) \in \mathcal{R}^{3\times 3}$ is the inertia tensors for the body about O. See [14] for more discussion on the properties and use of the spatial notation.

Serial Rigid Multibody Systems

This is the simplest case of a rigid multibody system. The system consists of n rigid bodies connected together by multiple dof hinges. The bodies are numbered 1 through n from tip to base. The term *outboard* (*inboard*) body refers to a body on the path towards the tip (base).

The generalized coordinates for the serial chain are the collection of the hinge configuration parameters. It is assumed that the kth hinge possesses $r_p(k)$ configuration degrees of freedom which are parameterized by the vector of configuration variables $\theta(k) \in \mathcal{R}^{r_p(k)}$ and that its $r_v(k)$ ($\leq r_p(k)$) motion dofs are parameterized by the generalized velocity vector $\beta(k) \in \mathcal{R}^{r_v(k)}$. The kinematical

equations which relate $\dot{\theta}(k)$ to $\beta(k)$ depend on the specific nature of the kth hinge. It is assumed for notational convenience that all the hinge constraints are homogeneous (i.e., catastatic). $H(k)$ is defined such that $H^*(k) \in \mathcal{R}^{6\times r_v(k)}$ is the hinge map matrix for the kth hinge. Its columns span the space of permissible relative spatial velocities across the hinge. The number of overall motion dofs for the serial chain are given by $\mathcal{N} \triangleq \Sigma_{k=1}^{n} r_v(k)$ for the chain. The state of the multibody system is defined by the collection of $[\theta(\cdot), \beta(\cdot)]$ pairs for all the hinges, and is assumed known.

Since each body is rigid, it suffices to develop the equations of motion for a body about a single reference point on the body, which is chosen here as the inboard hinge location $\mathcal{O}_k$ for the kth body (see Fig. 1). With $V(k) \in \mathcal{R}^6$ denoting the (inertial) spatial velocity, $\alpha(k) \in \mathcal{R}^6$ the (inertial) spatial acceleration, $f(k) \in \mathcal{R}^6$ the spatial interaction force and $T(k) \in \mathcal{R}^{r_v(k)}$ the generalized hinge force about $\mathcal{O}_k$ for the kth body, the following Newton-Euler recursive equations describe the equations of motion for the serial chain:

$$
\begin{cases}
\qquad V(n+1) = 0, \qquad \alpha(n+1) = 0 \\
\text{for } k = n \cdots 1 \\
\quad V(k) = \phi^*(k+1,k)V(k+1) + H^*(k)\beta(k) \\
\quad \alpha(k) = \phi^*(k+1,k)\alpha(k+1) + H^*(k)\dot{\beta}(k) + \alpha(k) \\
\text{end loop}
\end{cases}
$$

$$
\begin{cases}
\qquad f(0) = 0 \\
\text{for } k = 1 \cdots n \\
\quad f(k) = \phi(k+1,k)f(k-1) + M(k)\alpha(k) + b(k) \\
\quad T(k) = H(k)f(k) \\
\text{end loop}
\end{cases}
\tag{1}
$$

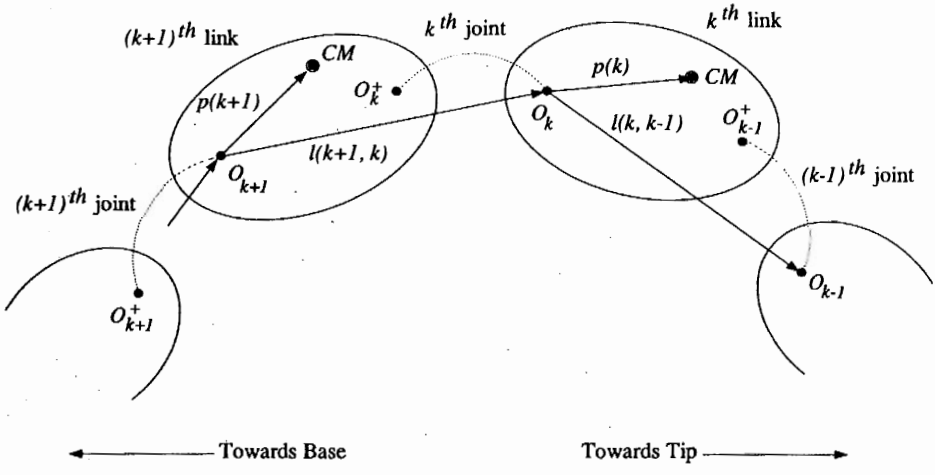

FIG. 1. Illustration of Links and Joints in a Serial Rigid Multibody System.

where the velocity dependent Coriolis and centrifugal acceleration term $a(k) \in \mathcal{R}^6$ is,

$$a(k) \triangleq \dot{\phi}^*(k+1,k)V(k+1) + \dot{H}^*(k)\beta(k) \tag{2}$$

The velocity dependent gyroscopic force term $b(k) \in \mathcal{R}^6$ is

$$b(k) \triangleq \dot{M}(k)V(k) - \dot{\phi}(p(k))M(k)V(k)$$

$\theta(k,k-1) \in \mathcal{R}^{6\times 6}$ denotes the transformation operator from $\mathcal{O}_{k-1}$ to $\mathcal{O}_k$. For more details on the derivation of these equations of motion, see [9,14].

The simplifying assumption is made that the tip force $f(0)$ is zero. Base mobility can easily be handled by attaching a full motion 6-dof hinge between the physical base and the inertial frame. For the inverse dynamics problem, the hinge accelerations $\dot{\beta}$ are known, and equation (1) represents an $O(\mathcal{N})$ computation involving a base-to-tip recursion to compute the velocities and accelerations, followed by a tip-to-base recursion to compute the hinge forces.

The equations of motion equation (1) can be expressed more concisely using spatial operators. In this notation, the $V(k)$'s, $\alpha(k)$'s etc. are viewed as components of vectors $V \in \mathcal{R}^{6n}$, $\alpha \in \mathcal{R}^{6n}$ etc. Then, equation (1) can be written in the following compact form:

$$\begin{aligned} V &= \mathcal{E}_\phi^* V + H^*\beta \\ \alpha &= \mathcal{E}_\phi^* \alpha + H^*\dot{\beta} + a \\ f &= \mathcal{E}_\phi f + M\alpha + b \\ T &= Hf \end{aligned} \tag{3}$$

where,

$$\mathcal{E}_\phi \triangleq \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \phi(2,1) & 0 & \cdots & 0 & 0 \\ 0 & \phi(3,2) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \phi(n,n-1) & 0 \end{bmatrix} \in \mathcal{R}^{6n\times 6n} \tag{4}$$

$$M \triangleq \begin{bmatrix} M(1) & 0 & \cdots & 0 \\ 0 & M(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & M(n) \end{bmatrix} \in \mathcal{R}^{6n\times 6n}$$

$$H \triangleq \begin{bmatrix} H(1) & 0 & \cdots & 0 \\ 0 & H(2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & H(n) \end{bmatrix} \in \mathcal{R}^{\mathcal{N}\times 6n} \tag{5}$$

However, since $\mathscr{E}_\phi$ is nilpotent ($\mathscr{E}_\phi^n = 0$),

$$\phi \triangleq (I - \mathscr{E}_\phi)^{-1} = I + \mathscr{E}_\phi + \mathscr{E}_\phi^2 + \cdots + \mathscr{E}_\phi^{n-1}$$

$$= \begin{bmatrix} I & 0 & \cdots & 0 \\ \phi(2,1) & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n,1) & \phi(n,2) & \cdots & I \end{bmatrix} \in \mathscr{R}^{6n \times 6n} \tag{6}$$

where,

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

Thus, equation (3) can be reexpressed as,

$$\begin{aligned} V &= \phi^* H^* \beta \\ \alpha &= \phi^*(H^* \dot{\beta} + a) \\ f &= \phi(M\alpha + b) = \phi M \phi^* H^* \dot{\beta} + \phi(M\phi^* a + b) \\ T &= Hf = H\phi M \phi^* H^* \dot{\beta} + H\phi(M\phi^* a + b) \\ &= \mathcal{M}\dot{\beta} + \mathscr{C}, \quad \text{where } \mathcal{M} \triangleq H\phi M \phi^* H^*, \quad \text{and} \quad C \triangleq H\phi(M\phi^* a + b) \end{aligned} \tag{7}$$

$\mathcal{M} \in \mathscr{R}^{N \times N}$ is the *mass matrix* for the serial chain and $\mathscr{C} \in \mathscr{R}^N$ contains the velocity dependent Coriolis, centrifugal and gyroscopic hinge forces. In the terminology of Kane's method [15], β are the *generalized speeds* and the elements of $\phi^* H^*$ are the *partial (spatial) velocities*. The operator expression $\mathcal{M} = H\phi M \phi^* H^*$ is denoted the *Newton-Euler Operator Factorization* of the mass matrix.

In terms of the rigid body transformation operator from the tip to the first hinge by $\phi(1,0)$, the tip spatial velocity, $V(0)$, is given by $V(0) = \phi^*(1,0)V(1)$. Thus,

$$V(0) = B^* \phi^* H^* \beta, \quad \text{where } B \triangleq \begin{bmatrix} \phi(1,0) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathscr{R}^{6n \times 6}$$

Thus, the Jacobian matrix $J \in \mathscr{R}^{6 \times N}$ which maps β to the tip velocity $V(0)$ is

$$J = B^* \phi^* H^*, \quad \text{and we have } V(0) = J\beta \tag{8}$$

$\mathscr{E}_\phi$, ϕ, H, and M are the first *spatial operators* encountered. Recursive dynamical algorithms can be derived easily by using the *state transition* properties [8] of the elements of spatial operators such as $\mathscr{E}_\phi$, ϕ etc. For instance, given a vector y, the evaluation of the matrix-vector product ϕy *does not* require an $O(n^2)$ matrix-vector product computation, and not even the explicit computation of the elements of ϕ. Rather, this product can be evaluated using an $O(n)$ recursive algorithm involving only the elements of $\mathscr{E}_\phi$ and y. This is precisely the cor-

respondence between the concise operator based high-level description of the equations of motion in equation (7) and the recursive algorithmic description in equation (1).

Spatially recursive $O(\mathcal{N})$ forward dynamics algorithms for serial chains have been developed in [3] based on the recognition of the close parallels between the structure of the multibody dynamics equations and the equations encountered in Kalman Filtering theory [16]. These parallels are used to obtain a new (square) Innovations operator factorization of the mass matrix $\mathcal{M}$ and an operator expression for its inverse, and form the basis for the recursive algorithms.

Tree Topology Systems

In this section, the dynamics of tree-topology rigid multibody systems are discussed. A tree topology system is viewed here as a set of component serial chains (referred to as *branches*), coupled together via hinges at their inboard *terminal* bodies. The total number of branches is denoted ℓ. The index for the branches ranges from $1 \cdots \ell$, and consistent with the body numbering scheme in the previous section, the inboard branches are assigned indices larger than those for the outboard ones. The *inboard function* $i(k)$ is defined as the index of the *direct predecessor* branch, i.e., the inboard branch to which the kth branch is directly connected. The jth branch is referred to as a *predecessor* of the kth branch if it belongs on the unique path from the kth branch to the base, i.e., if $i^p(k) = j$ for some integer $p > 0$, where $i^p(\cdot) = i \circ i \circ \cdots \circ i(\cdot)$ denotes a p-times function composition. Figure 2 illustrates the decomposition of a tree system into branches as well as a sample branch numbering scheme for the system.

The notation developed earlier for serial chains is used to describe the branches in the tree system. A subscript serves to identify the specific branch in the system. Thus n_j and $\mathcal{N}_j$ denote the number of bodies and the number of motion dofs respectively, while $V_j \in \mathcal{R}^{6n_j}$, and M_j, $\mathcal{C}_{\phi j}$, ϕ_j etc. in $\mathcal{R}^{6n_j \times 6n_j}$ denote the spatial vector, spatial inertias etc. for the jth branch. A body/hinge is identified by the index of the branch it is on, plus its location within the branch. For instance, the k_j body is the kth body on the jth branch, and $V(k_j)$ (or $V_j(k)$) denotes the spatial velocity of the kth body of the jth branch at its inboard hinge location $\mathcal{O}(k_j)$. The overall stacked spatial velocity, acceleration etc. vectors for the tree are once again denoted V, α, f etc. with $V \triangleq [V_1^* \cdots V_\ell^*]^* \in \mathcal{R}^{6n}$ etc. The total number of bodies n, and the total number of motion dofs $\mathcal{N}$ for the tree multibody system are now given by

$$n \triangleq \sum_{j=1}^{\ell} n_j \quad \text{and} \quad \mathcal{N} \triangleq \sum_{j=1}^{\ell} \mathcal{N}_j \tag{9}$$

Note that when the jth branch is the direct predecessor of the kth branch, i.e., $j = i(k)$, the hinge connecting them is regarded as the n_kth hinge (the last hinge) on the kth branch and describes the attachment to body 1_j on the jth branch. The transformation operator from the n_kth hinge to the 1_jth hinge is denoted $\phi(1_j, n_k)$. The spatial operator $\mathcal{C}_\phi \in \mathcal{R}^{6n \times 6n}$ for the tree system is defined in terms of its block matrix elements below. For $j, k \in 1 \cdots \ell$, the (j,k)th block element

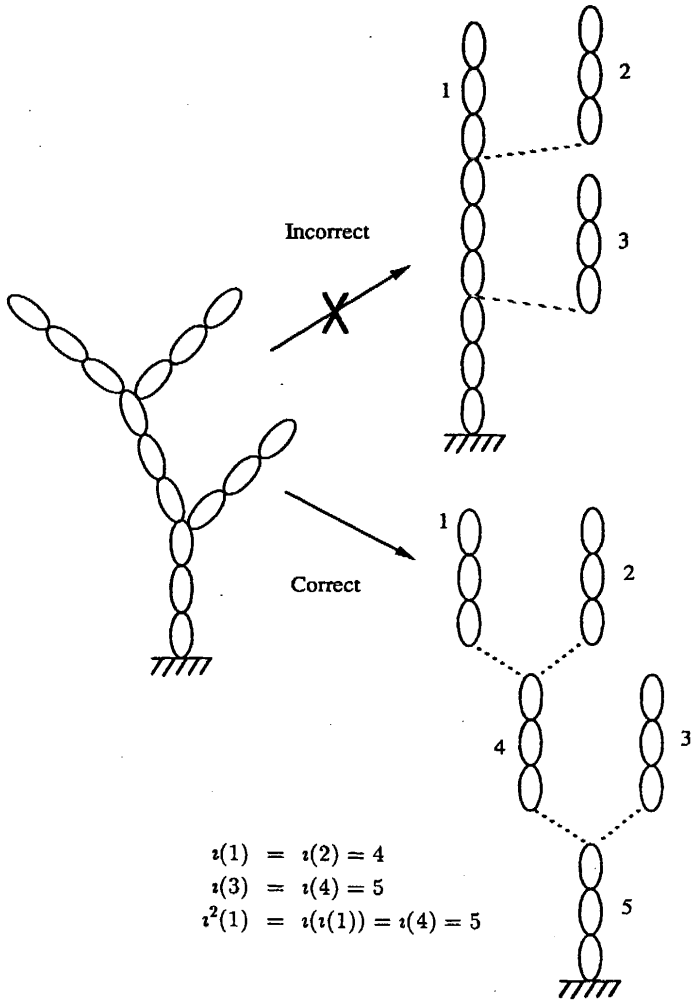

FIG. 2. Illustration of Decomposition into Branches for a Tree System.

$\mathscr{E}_\phi(j,k) \in \mathscr{R}^{6n_j \times 6n_k}$ of $\mathscr{E}_\phi$ is defined as follows:

$$\mathscr{E}_\phi(j,k) = \begin{cases} \mathscr{E}_{\phi_j} & \text{for } j = k \\ \mathscr{E}_{\phi_{j,k}} \triangleq \begin{bmatrix} 0 & \cdots & 0 & \phi(1_j, n_k) \\ 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} & \text{for } j = i(k), \text{ i.e. if } j = k\text{'s direct predecessor branch} \\ 0 & \text{for } j \neq i(k), \text{ i.e. if } j \neq k\text{'s direct predecessor branch} \end{cases} \quad (10)$$

In this paper 0 denotes a zero matrix whose dimension is apparent from the context. As a consequence of the numbering scheme used here, for $j < k$, the jth branch cannot be a predecessor to the kth branch and thus the (j,k)th block element, $\mathscr{E}_\phi(j,k) = 0$. Thus $\mathscr{E}_\phi$ is a strictly lower triangular matrix. The analogs of equation (3) are as follows:

$$\begin{aligned} V &= \mathscr{E}_\phi^* V + H^* \beta \\ \alpha &= \mathscr{E}_\phi^* \alpha + H^* \dot{\beta} + a \\ f &= \mathscr{E}_\phi f + M\alpha + b \\ T &= Hf \end{aligned} \tag{11}$$

Once again (analogous to equation (6)), $\mathscr{E}_\phi$ is nilpotent ($\mathscr{E}_\phi^n = 0$) and the operator ϕ can be defined as follows:

$$\phi \triangleq (I - \mathscr{E}_\phi)^{-1} = I + \mathscr{E}_\phi + \mathscr{E}_\phi^2 + \cdots + \mathscr{E}_\phi^{n-1} \tag{12}$$

It is easy to verify that the (j,k)th block element $\phi(j,k) \in \mathscr{R}^{6n_j \times 6n_k}$ of ϕ is given below:

$$\phi(j,k) = \begin{cases} \phi_j \triangleq (I - \mathscr{E}_{\phi_j})^{-1} & \text{for } j = k \\ \phi_{j,k} & \text{if } \exists\, p > 0 \colon j = i^p(k), \text{ i.e., if } j \text{ is a predecessor branch of } k \\ 0 & \text{if } j \neq i^p(k) \quad \forall\, p > 0, \text{ i.e., if } j \text{ is } \textit{not} \text{ a predecessor branch of } k \end{cases} \tag{13}$$

where components of the matrix $\phi_{j,k}$ are defined as

$$\phi_{j,k}(m,l) = \phi(m_j, l_k) \triangleq \phi(m_j, m_j - 1)\phi(m_j - 1, m_j - 2)\cdots\phi(l_k + 1, l_k)$$

for $m \in 1\cdots n_j$ and $l \in 1\cdots n_k$. $\phi(m_j, l_k)$ is the transformation operator from hinge l_k (on the kth branch) to hinge m_j (on the jth branch) and is a generalization of the transformation operator $\phi(i,j)$ in equation (6) for serial chains. It is formed by sequentially composing all the individual transformation operators that lie on the *unique* path joining the two hinges. The numbering scheme used here ensures that ϕ will be a lower triangular matrix. The structure of the $\mathscr{E}_\phi$ and ϕ operators for the example in Fig. 2 is given below:

$$\mathscr{E}_\phi = \begin{bmatrix} \mathscr{E}_{\phi_1} & 0 & 0 & 0 & 0 \\ 0 & \mathscr{E}_{\phi_2} & 0 & 0 & 0 \\ 0 & 0 & \mathscr{E}_{\phi_3} & 0 & 0 \\ \mathscr{E}_{\phi_{4,1}} & \mathscr{E}_{\phi_{4,2}} & 0 & \mathscr{E}_{\phi_4} & 0 \\ 0 & 0 & \mathscr{E}_{\phi_{5,3}} & \mathscr{E}_{\phi_{5,4}} & \mathscr{E}_{\phi_5} \end{bmatrix}, \quad \text{and} \quad \phi = \begin{bmatrix} \phi_1 & 0 & 0 & 0 & 0 \\ 0 & \phi_2 & 0 & 0 & 0 \\ 0 & 0 & \phi_3 & 0 & 0 \\ \phi_{4,1} & \phi_{4,2} & 0 & \phi_4 & 0 \\ \phi_{5,1} & \phi_{5,2} & \phi_{5,3} & \phi_{5,4} & \phi_5 \end{bmatrix}$$

The operator ϕ has state transition properties analogous to the ϕ for serial chains, and as a consequence, it can be used for high-level and concise description and analysis of the dynamics of tree topology systems (as in equation (14) below). However, from the computational perspective, equations involving these

operators can always be directly mapped into very efficient and recursive computational algorithms. From equation (11) and equation (12) it follows that,

$$
\begin{aligned}
V &= \phi^* H^* \beta \\
\alpha &= \phi^* (H^* \dot{\beta} + a) \\
f &= \phi(M\alpha + b) = \phi M \phi^* H^* \dot{\beta} + \phi(M\phi^* a + b) \\
T &= Hf = H\phi M \phi^* H^* \dot{\beta} + H\phi(M\phi^* a + b) \\
&= \mathcal{M}\dot{\beta} + \mathcal{C}, \qquad \text{where } \mathcal{M} \triangleq H\phi M \phi^* H^*, \text{ and } \mathcal{C} \triangleq H\phi(M\phi^* a + b)
\end{aligned} \tag{14}
$$

$\mathcal{M} \in \mathcal{R}^{N \times N}$ denotes the mass matrix for the tree system, and $\mathcal{C} \in \mathcal{R}^N$ is the vector of velocity dependent nonlinear Coriolis type terms. While neither $\mathcal{M}$ nor $\mathcal{C}$ are typically available, their explicit computation is not required either for solving the forward dynamics of the system. Since the complexity of the forward dynamics algorithm depends primarily on handling $\mathcal{M}$ and not $\mathcal{C}$, we assume here for simplicity that $\mathcal{C}$ is explicitly computed from the state information. Defining $\overline{T} \triangleq T - \mathcal{C}$, the forward dynamics then requires the solution of

$$\mathcal{M}\dot{\beta} = \overline{T} \tag{15}$$

for the hinge accelerations $\dot{\beta}$.

It is noteworthy that the spatial operator expressions for the mass matrix $\mathcal{M}$ and the Coriolis vector $\mathcal{C}$ in equation (14) are identical to the corresponding ones for serial chains in equation (7). Moreover, key properties of the operators remain uneffected. Thus, important operator results for the alternative (square) innovations factorization and the inversion of the mass matrix for serial chain systems in [9] also hold for tree-topology systems. Indeed, with only minor extensions, even the $O(N)$ recursive algorithms for the dynamics of serial chain systems extend directly to tree-topology systems. The discussion of these operator results and algorithms is postponed until after the dynamics of closed-chain systems have been discussed in the next section.

However, before proceeding to closed topology systems, an expression is derived for the tree-topology Jacobian operator. Given n_C points, denoted C_k's on the tree, the Jacobian operator $J \in \mathcal{R}^{6n_C \times N}$ defines the mapping between β and $\hat{V}$, i.e., $\hat{V} = J\beta$, where $\hat{V} \in \mathcal{R}^{6n_C}$ denotes the vector of spatial velocities at these points. If C_k is on body m_j, then the spatial velocity at C_k is given by

$$\hat{V}(k) = \phi^*[\mathcal{O}(m_j), C_k] V(m_j) \tag{16}$$

with $\phi(\mathcal{O}(m_j), C_k)$ denoting the rigid body transformation operator from C_k to the point $\mathcal{O}(m_j)$. With the block elements of $B \in \mathcal{R}^{6n \times 6n_C}$ defined as

$$B(m_j, k) = \begin{cases} \phi[\mathcal{O}(m_j), C_k] & \text{if } C_k \in m_j\text{th body} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } k = 1 \cdots n_C \tag{17}$$

it follows from equation (16) that

$$\hat{V} = B^* V = B^* \phi^* H^* \beta, \qquad \text{i.e., } J = B^* \phi^* H^* \tag{18}$$

This gives an expression for the desired Jacobian operator that will be used below to deal with the loop closure constraints for closed topology systems.

Closed Topology Systems

This section describes equations of motion for closed topology multibody systems with time-varying topologies, constraints and constituent bodies. The closed-chain system consists of:

(a) the *primary* tree-topology subsystem, consisting of the major part whose topology and constituent bodies are the least time-variant.
(b) The *secondary* tree-topology subsystem, which is much smaller, and consists of the part whose topology and constituent bodies do change from time to time.
(c) The set of *closure constraints* and/or boundary conditions between/within the primary and secondary systems.

Note that the subsystems described above are in an order of increasing time-variation. As an example, take the case of multiple robot manipulators interacting with each other and the environment to perform complex tasks. In this context, the manipulators by themselves should be assigned to the primary system since their internal structure does not vary with time. On the other hand, the task objects and the tools vary from task to task and are assigned to the secondary system. The closure constraints between the primary and secondary subsystems characterize those arising from task related grasping, mating, tool operation etc., and belong to the last category.

This decomposition of the closed topology system is a departure from the more traditional approach (see [10, 11]) of forming a spanning tree for the full system and computing the constraint forces at the points of closure. In these latter approaches, even small changes in the original system typically require whole new spanning trees for the system. This disallows any algorithmic optimization, and the algorithms are also not amenable to coping with time-varying systems. When the application system is "time-invariant", such as a spacecraft, there may be no secondary subsystem at all, and in the absence of closed loops, the set of constraints will also be empty.

The equations of motion for tree topology systems derived in equation (15) will be used to describe the dynamics of the tree components of both the primary and secondary systems, with the subscripts "P" and "S" being used to identify the two subsystems. Thus the dynamics of the tree part of the two systems are described by

$$\overline{T}_P = H_P \phi_P M_P \phi_P^* H_P^* \dot{\beta}_P = \mathcal{M}_P \dot{\beta}_P, \qquad \text{and} \quad \overline{T}_S = H_S \phi_S M_S \phi_S^* H_S^* \dot{\beta}_S = \mathcal{M}_S \dot{\beta}_S \quad (19)$$

$\mathcal{M}_P$ and $\mathcal{M}_S$ denote the mass matrices, β_P and β_S the motion dof parameter vectors, $\overline{T}_P$ and $\overline{T}_S$ the bias-free internal hinge forces for the primary and secondary subsystems respectively.

Collect together the internal loop points of closure with the points of closure coupling the two systems, and denote their number by ζ_P and ζ_S for the primary and secondary systems respectively. Let $\hat{V}_P \in \mathcal{R}^{6\zeta_P}$ and $\hat{V}_S \in \mathcal{R}^{6\zeta_S}$ denote the vec-

tors of spatial velocities at these points of closure for the two systems. Following the discussion leading up to equation (18), $J_P = B_P^* \phi_P^* H_P^*$ and $J_S = B_S^* \phi_S^* H_S^*$ are the Jacobians to these points of closure for the two systems respectively. Thus $\hat{V}_P = J_P \beta_P$ and $\hat{V}_S = J_S \beta_S$. The kinematical constraints due to the existence of internal closed loops within the primary and secondary systems are characterized by constraint matrices Q_P and Q_S and lead to kinematical constraint equations of the form:

$$Q_P \hat{V}_P = \hat{U}_P \quad \text{and} \quad Q_S \hat{V}_S = \hat{U}_S$$

The coupling together of the primary and secondary systems is characterized by the constraint matrices $\hat{Q}_P$ and $\hat{Q}_S$ and leads to a kinematical constraint equation of the form:

$$\hat{Q}_P \hat{V}_P + \hat{Q}_S \hat{V}_S = \hat{U}_C$$

Define

$$A_P \triangleq \begin{pmatrix} \hat{Q}_P \\ Q_P \\ 0 \end{pmatrix}, \qquad A_S \triangleq \begin{pmatrix} \hat{Q}_S \\ 0 \\ Q_S \end{pmatrix}, \qquad \text{and} \quad A \triangleq [A_P \; A_S]$$

The closure constraints can be collectively expressed as:

$$A \begin{pmatrix} \hat{V}_P \\ \hat{V}_S \end{pmatrix} = [A_P \; A_S] \begin{pmatrix} J_P & 0 \\ 0 & J_S \end{pmatrix} \begin{pmatrix} \beta_P \\ \beta_S \end{pmatrix} = [A_P J_P \; A_S J_S] \begin{pmatrix} \beta_P \\ \beta_S \end{pmatrix} = \begin{pmatrix} \hat{U}_C \\ \hat{U}_P \\ \hat{U}_S \end{pmatrix} \triangleq \hat{U} \qquad (20)$$

It is assumed here onwards that $[A_P J_P \; A_S J_S]$ is of full row rank, and its rank is $\mathcal{N}_E$. The overall number of motion dofs of the closed chain system is given by $\mathcal{N}_C \triangleq \mathcal{N} + \mathcal{N}_S - \mathcal{N}_E$. While not required for the purpose here, equation (20) can be used, whenever necessary, to find the $\mathcal{N}_C$ dimensional minimal set of generalized velocities for the system. Based on the principle of virtual work, equation (20) implies that the closure constraint forces are

$$\begin{pmatrix} J_P^* A_P^* \\ J_S^* A_S^* \end{pmatrix} \hat{f} \qquad (21)$$

for some $\hat{f} \in \mathcal{R}^{\mathcal{N}_E}$. Together, the dynamical equations of motion for the primary and secondary systems equation (7) and equation (21), as well as the constraint equation in equation (20), lead to the following equations of motion for the closed chain system follows:

$$\begin{pmatrix} \mathcal{M}_P & 0 & J_P^* A_P^* \\ 0 & \mathcal{M}_S & J_S^* A_S^* \\ A_P J_P & A_S J_S & 0 \end{pmatrix} \begin{pmatrix} \dot{\beta}_P \\ \dot{\beta}_S \\ \hat{f} \end{pmatrix} = \begin{pmatrix} \overline{T}_P \\ \overline{T}_S \\ U \end{pmatrix}, \qquad \text{where } U \triangleq \dot{\hat{U}} - [(\dot{A}_P J_P)(\dot{A}_S J_S)] \begin{pmatrix} V_P \\ V_S \end{pmatrix} \qquad (22)$$

Premultiplication of both sides of the above equation by the matrix

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ -A_P J_P \mathcal{M}_P^{-1} & -A_S J_S \mathcal{M}_S^{-1} & I \end{pmatrix}$$

leads to

$$\begin{pmatrix} \mathcal{M}_P & 0 & J_P^* A_P^* \\ 0 & \mathcal{M}_S & J_S^* A_S^* \\ 0 & 0 & -[A_P \Lambda_P A_P^* + A_S \Lambda_S A_S^*] \end{pmatrix} \begin{pmatrix} \dot{\beta}_P \\ \dot{\beta}_S \\ \hat{f} \end{pmatrix} = \begin{pmatrix} \overline{T}_P \\ \overline{T}_S \\ U - [A_P J_P \mathcal{M}_P^{-1} \overline{T}_P + A_S J_S \mathcal{M}_S^{-1} \overline{T}_S] \end{pmatrix} \quad (23)$$

where

$$\Lambda_P \triangleq J_P \mathcal{M}_P^{-1} J_P^*, \quad \text{and} \quad \Lambda_S \triangleq J_S \mathcal{M}_S^{-1} J_S^*$$

Physically, $\Lambda_P \in \mathcal{R}^{6\zeta_P \times 6\zeta_P}$ and $\Lambda_S \in \mathcal{R}^{6\zeta_S \times 6\zeta_S}$ are the effective "admittances" of the primary and secondary systems reflected to the points of closure.

When there is no secondary system, the equations of motion in equation (22) are given by

$$\begin{pmatrix} \mathcal{M}_P & J_P^* A_P^* \\ A_P J_P & 0 \end{pmatrix} \begin{pmatrix} \dot{\beta}_P \\ \hat{f} \end{pmatrix} = \begin{pmatrix} \overline{T}_P \\ U \end{pmatrix}, \quad \text{where } U \triangleq \dot{\hat{U}} - (\dot{A}_P J_P) V_P$$

The corresponding form of equation (23) is given by:

$$\Rightarrow \begin{pmatrix} \mathcal{M}_P & J_P^* A_P^* \\ 0 & -A_P \Lambda_P A_P^* \end{pmatrix} \begin{pmatrix} \dot{\beta}_P \\ \hat{f} \end{pmatrix} = \begin{pmatrix} \overline{T}_P \\ U - A_P J_P \mathcal{M}_P^{-1} \overline{T}_P \end{pmatrix} \quad (24)$$

Simplifications for some special cases types of closed chain systems are described below:

- When the hinge constraints coupling the primary and secondary systems are only on the relative spatial velocity across the hinges coupling them together, an appropriate reordering of the elements of $\hat{V}$ will result in $\hat{Q}_P = -\hat{Q}_S$. Furthermore, if no relative motion is permitted across the hinge, i.e., there is rigid rather than loose coupling, then in fact $\hat{Q}_P = I$ and $\hat{Q}_S = I$. When this feature holds for only some of the hinges, only the corresponding rows of Q_P and Q_S have this special structure.
- When the secondary system has no internal actuators or source of generalized forces, then $T_S = 0$.
- If the secondary system is a free rigid body with no internal degrees of freedom, then the motion generalized coordinates vector β_S is of dimension 6 and is just the spatial velocity of the body.

Forward Dynamics of Closed Chain Systems

This section describes a recursive algorithm for solving the forward dynamics of closed chain rigid multibody systems which does not require the explicit computation of the system mass matrix.

From the equations of motion for closed chain systems given by equation (23), the forward dynamics problem can be solved by the following sequence of steps:

(A) Solve $\mathcal{M}_P \dot{\beta}_P^f = \overline{T}_P$ for $\dot{\beta}_P^f$ Solve $\mathcal{M}_S \dot{\beta}_S^f = \overline{T}_S$ for $\dot{\beta}_S^f$

(B) Compute $\alpha_P^f = J_P \dot{\beta}_P^f$ Compute $\alpha_S^f = J_S \dot{\beta}_S^f$

(C) Compute $\Lambda_P = J_P \mathcal{M}_P^{-1} J_P^*$ Compute $\Lambda_S = J_S \mathcal{M}_S^{-1} J_S^*$

(D) Solve $[A_P \Lambda_P A_P^* + A_S \Lambda_S A_S^*]\hat{f} = (A_P \alpha_P^f + A_S \alpha_S^f) - U$ for $\hat{f}$

(E) Solve $\mathcal{M}_P \dot{\beta}_P^\delta = -J_P^* A_P^* \hat{f}$ for $\dot{\beta}_P^\delta$ Solve $\mathcal{M}_S \dot{\beta}_S^\delta = -J_S^* A_S^* \hat{f}$ for $\dot{\beta}_S^\delta$

(F) $\dot{\beta}_P = \dot{\beta}_P^f + \dot{\beta}_P^\delta$ $\dot{\beta}_S = \dot{\beta}_S^f + \dot{\beta}_S^\delta$

The interpretation of each of the steps is as follows. STEP (A) solves for the "free" hinge accelerations $\dot{\beta}_P^f$ and $\dot{\beta}_S^f$ of the primary and secondary systems assuming that there are no closure constraints on the system. Note that this step is equivalent to computing the forward dynamics of the primary/secondary tree-topology systems. STEP (B) computes the corresponding "free" tip spatial accelerations α_P^f and α_S^f for the systems. STEP (C) computes the effective admittances of the primary and secondary systems at the points of closure. In STEP (D), the constraint forces at the points of closure are computed, and they are used in STEP (E) to compute the "correction" hinge accelerations $\dot{\beta}_P^\delta$ and $\dot{\beta}_S^\delta$ for the two systems. Combining the free and correction hinge accelerations in STEP (F) gives the true hinge accelerations for the two systems.

It is now easy to see the use of the decomposition into primary and secondary systems in the development of dynamics algorithms which are responsive and adaptable to time-varying systems. The forward dynamics procedure above involves a sequence of decoupled steps for each of the primary and secondary system dynamics, and one step in which they come together when the constraint forces are computed. Being structurally time-invariant, it is possible to put in place optimized algorithms for the dynamics of the primary system. Changes in the secondary system, which is typically of small dimension, effect only the relatively smaller number of computations involving the steps in the right half column. Changes in the closure constraints only alters the constraint matrix A, and thus only STEP D is affected, while the computations for the primary and secondary systems remain unaffected. Thus, changes to the secondary system or the constraints on the system require very modest changes to reconfigure the algorithm to respond to these changes.

Recursive algorithms for each of the above steps are now described. It follows from the symmetry between the steps in the two columns that the recursive algorithms for the primary system are directly applicable to the secondary system

as well. Therefore, the explicit use of the subscripts identifying the primary/secondary systems are dropped here onwards (except for STEP (D)).

STEP (A) Solve $\mathcal{M}\dot{B}^f = \bar{T}$. (Forward Dynamics of a Tree Topology System)

Step (A) is equivalent to solving the forward dynamics of a tree topology system. Here, an $O(N)$ recursive algorithm for this solution is developed. This algorithm is based on a new factorization of the mass matrix $\mathcal{M}$ in terms of square factors in contrast to the earlier non-square factorization in equation (14). This square factorization is then used to obtain an explicit expression for $\mathcal{M}^{-1}$.

The *articulated body inertia matrix* P is defined as the solution to the following equation:

$$M = P - \mathcal{E}_\phi[P - PH^*(HPH^*)^{-1}HP]\mathcal{E}_\phi^* \tag{25}$$

P is *block diagonal* and the elements on the diagonal (denoted $P(k_j)$) can be obtained using a recursive algorithm described in equation (39) in Appendix A. Physically, $P(k_j)$ is the *articulated body inertia* as seen at the k_jth hinge, i.e., it is the effective inertia of all the links outboard from the k_jth hinge assuming that the hinge forces at all the outboard hinges are zero.

For the subsequent development, it is convenient to define

$$D \triangleq HPH^*, \qquad G \triangleq PH^*D^{-1}, \qquad K \triangleq \mathcal{E}_\phi G$$
$$\tau \triangleq GH, \qquad \bar{\tau} \triangleq I - \tau, \qquad \mathcal{E}_\psi \triangleq \mathcal{E}_\phi\bar{\tau} \tag{26}$$

Note that D, G, τ and $\bar{\tau}$ are all block diagonal. The structure of $\mathcal{E}_\psi$ is identical to that of $\mathcal{E}_\phi$ with its elements being given by

$$\psi(k_j, k_j - 1) \triangleq \phi(k_j, k_j - 1)\bar{\tau}(k_j - 1)$$

$\mathcal{E}_\psi$ is also nilpotent ($\mathcal{E}_\psi^n = 0$), and analogous to ϕ, ψ is defined as

$$\psi \triangleq (I - \mathcal{E}_\psi)^{-1} = I + \mathcal{E}_\psi + \mathcal{E}_\psi^2 + \cdots + \mathcal{E}_\psi^{n-1} \tag{27}$$

The structure of ψ is very similar to that of ϕ and it also possesses the state transition properties which are used to develop recursive algorithms. ϕ may be viewed as the transformation operator for composite bodies (i.e., as if all the hinges are locked), while ψ is the transformation operator for articulated bodies (i.e., as if all the hinge forces were zero). The following lemma yields a square factorization of $\mathcal{M}$.

Lemma 1: The mass matrix $\mathcal{M}$ has the following factorization:

$$\mathcal{M} = [I + H\phi K]D[I + H\phi K]^*, \tag{28}$$

Proof: See Appendix B. ∎

Note that the factor $[I + H\phi K]$ is square and block lower triangular, while D is block diagonal. Thus the factorization in equation (28) may be thought of as an explicit block LDU factorization of the mass matrix. This factorization is also known as the *innovations factorization* because of its relationship to the innovations approach to filtering and prediction theory (see [17]).
The following lemma gives the explicit form for the inverse of $[I + H\phi K]$.

Lemma 2:

$$[I + H\phi K]^{-1} = [I - H\psi K] \tag{29}$$

Proof: See Appendix B. ∎

Combination of Lemma 1 and Lemma 2 leads to the following form for the inverse of the mass matrix.

Lemma 3:

$$\mathcal{M}^{-1} = [I - H\psi K]^* D^{-1}[I - H\psi K] \tag{30}$$ ∎

Note that the factor $[I - H\psi K]$ is square and block lower triangular, while D^{-1} is block diagonal. Thus the factorization in equation (29) may be thought of as an explicit block LDU factorization of the mass matrix inverse. Thus,

$$\dot{\beta}^f = \mathcal{M}^{-1}\overline{T} = [I - H\psi K]^* D^{-1}[I - H\psi K]\overline{T} \tag{31}$$

The $O(\mathcal{N})$ recursive computation of the expression on the right is given in equation (40) in Appendix A.

STEP (B) Compute $\alpha^f = J\dot{B}^f$

From equation (18), $\alpha^f = B^*\tilde{\alpha}^f$, where

$$\tilde{\alpha}^f \triangleq \phi^* H^* \dot{\beta}^f \tag{32}$$

However,

Lemma 4:

$$(I - H\psi K)H\phi = H\psi \tag{33}$$ ∎

Proof: See Appendix B. ∎

Thus, use of equation (30) and the above lemma in equation (32),

$$\tilde{\alpha}^f = \phi^* H^* [I - H\psi K]^* D^{-1}[I - H\psi K]\overline{T} = \psi^* H^* D^{-1}[I - H\psi K]\overline{T}$$

Compare this with equation (31) to see that $\tilde{\alpha}^f$ can be evaluated as an intermediate quantity in the $O(\mathcal{N})$ recursive algorithm for computing $\dot{\beta}^f$ described in STEP (A).

STEP (C) Compute $\Lambda = J\mathcal{M}^{-1}J^*$

Use of equation (18) and equation (30) implies

$$\begin{aligned}\Lambda &= \{[I - H\psi K]H\phi B\}^* D^{-1}\{[I - H\psi K]H\phi B\} \\ &= B^*\psi^* H^* D^{-1} H\psi B = B^*\Omega B, \qquad \text{where } \Omega \triangleq \psi^* H^* D^{-1} H\psi \end{aligned} \tag{34}$$

Here equation (33) has been used to simplify the above expression. A recursive $O(N)$ procedure for the computation of Ω is given in equation (43) in Appendix A. Note that without the simplification resulting from the use of equation (33), the computation of Λ would be an $O(N^3)$ process.

STEP (D) Solve $[A_P \Lambda A_P^* + A_S \Lambda_S A_S^*]\hat{f} = (A_P \alpha_P^f + A_S \alpha_S^f) - U$ for $\hat{f}$

Now,

$$\hat{f} = [A_P \Lambda A_P^* + A_S \Lambda_S A_S^*]^{-1}[(A_P \alpha_P^f + A_S \alpha_S^f) - U] \tag{35}$$

In this form this step is of $O(N_E^3)$ complexity. However, when $(A_P \Lambda_P A_P^*)$ is invertible, there is an alternative expression for $\hat{f}$. This is obtained by reexpressing equation (23) as:

$$\begin{pmatrix} \mathcal{M}_P & 0 & J_P^* A_P^* \\ 0 & \mathcal{M}_S & J_S^* A_S^* \\ 0 & A_S J_S & -A_P \Lambda_P A_P^* \end{pmatrix} \begin{pmatrix} \dot{\beta} \\ \dot{\beta}_S \\ \hat{f} \end{pmatrix} = \begin{pmatrix} \overline{T}_P \\ \overline{T}_S \\ U - A_P \alpha_P^f \end{pmatrix}$$

Premultiply both sides by

$$\begin{pmatrix} I & 0 & 0 \\ 0 & I & J_S A_S^* (A_P \Lambda_P A_P^*)^{-1} \\ 0 & 0 & I \end{pmatrix}$$

to get

$$\begin{pmatrix} \mathcal{M}_P & 0 & J_P^* A_P^* \\ 0 & \mathcal{M}_S + J_S^* A_S^* (A_P \Lambda A_P^*)^{-1} A_S J_S & 0 \\ 0 & A_S J_S & -A_P \Lambda_P A_P^* \end{pmatrix} \begin{pmatrix} \dot{\beta} \\ \dot{\beta}_S \\ \hat{f} \end{pmatrix}$$

$$= \begin{pmatrix} \overline{T}_P \\ \overline{T}_S + J_S^* A_S^* (A_P \Lambda_P A_P^*)^{-1}[U - A_P \alpha_P^f] \\ U - A_P \alpha_P^f \end{pmatrix}$$

From the above equation it follows that

$$\begin{aligned} \dot{\beta}_S &= [\mathcal{M}_S + J_S^* A_S^* (A_P \Lambda_P A_P^*)^{-1} A_S J_S]^{-1}[\overline{T}_S - J_S^* A_S^* (A_P \Lambda A_P^*)^{-1}(A_P \alpha_P^f - U)] \\ \hat{f} &= (A_P \Lambda A_P^*)^{-1}[(A_P \alpha_P^f + A_S J_S \dot{\beta}_S) - U] \\ &= (A_P \Lambda_P A_P^*)^{-1}[(A_P \alpha_P^f + A_S \alpha_S) - U], \qquad \text{where } \alpha_S = J_S \dot{\beta}_S \end{aligned}$$

Note the similarity between the forms of equation (35) and the above equation for $\hat{f}$. The computational cost of the above operation is a combination of the cost of inverting $A_P \Lambda_P A_P^*$, and the $O(N_S^3)$ step of solving a square linear system

of equations of size $\mathcal{N}_S$. The cost of inverting $A_P\Lambda_P A_P^*$ depends on its structure: its sparsity reflects the degree of coupling between the closed loops in the system. The cost is typically much less than the worst case of $O(\mathcal{N}_E^3)$. In many application domains such as robotics, $A_P\Lambda_P A_P^*$ is in fact block diagonal and is thus invertible in $O(\mathcal{N}_E)$ steps [8]. In addition, for most applications $\mathcal{N}_S \ll \mathcal{N}_E$, and this new formulation can lead to considerable computational savings.
The inverse of $[A_P\Lambda_P A_P^* + A_S\Lambda_S A_S^*]$ will not exist if $[A_P J_P \; A_S J_S]$ is not of full rank, i.e., the configuration is such that the number of motion dofs for the system have changed. It is therefore necessary to reformulate the constraint equation (20) so as to preserve the full rank property. Such changes of rank can occur at kinematically singular configurations.

STEP (E) Compute $\beta^\delta = -\mathcal{M}^{-1}J^*A^*\hat{f}$

From equation (30) and equation (18),

$$\beta^\delta = -[I - H\psi K]^*D^{-1}[I - H\psi K]H\phi BA^*\hat{f}$$

Use of Lemma 4 in this leads to

$$\beta^\delta = -[I - H\psi K]^*D^{-1}H\psi BA^*\hat{f} \tag{36}$$

The recursive $O(\mathcal{N})$ implementation of the above step is given in equation (44) in Appendix A.

The overall complexity of this spatially recursive forward dynamics algorithm ranges between $O(\mathcal{N} + \mathcal{N}_S) + O(\mathcal{N}_E^3)$ for the worst case and $O(\mathcal{N} + \mathcal{N}_S) + O(\mathcal{N}_E) + O(\mathcal{N}_S^3)$ in the best case.

By treating the primary and secondary system as one overall system, which amounts to defining the quantities $\psi \triangleq \text{diag}(\psi_P, \psi_S)$, $H \triangleq \text{diag}(H_P, H_S)$ etc., and using the above results, for $U = 0$ the overall closed topology forward dynamics algorithm can be restated in the following form:

$$\dot{\beta} = [I - H\psi K]^*D^{-1/2}[I - b(A\Lambda A^*)^{-1}b^*]D^{-1/2}[I - H\psi K]\bar{T},$$
$$\text{where } b \triangleq D^{-1/2}H\psi BA^* \tag{37}$$

Note that when there are no closed loops in the overall system, $A = 0$, and the middle term reduces to I. Hence, the forward dynamics of tree topology systems in equation (31) are recovered.

Conclusions

The algorithms developed here are suitable for multibody systems that have time-varying topology as well as changing constituent bodies and constraints. For the sake of clarity, the focus of much of the paper is on multibody systems with rigid links.

This paper describes the use of the *spatial operator algebra* to easily develop these algorithms. Based on the rate of time-variation, a multibody system is par-

titioned into a primary subsystem, a secondary subsystem and the set of closure constraints. Spatial operators are used to develop operator factorizations and inversion of the mass matrix. These factorizations lead directly to efficient computational algorithms for the dynamics of the multibody system. The algorithm consists of parallel paths involving the tree-topology primary and secondary systems respectively. The two paths come together at one point to compute the constraint forces. The algorithm can be adapted to time-varying topology and changes in constraints or constituent bodies since only localized and relatively easy modifications to the algorithm are required.

The algorithm does not require the computation of the mass matrix, and its computational complexity is linear in the number of degrees of freedom, $(\mathcal{N}_P + \mathcal{N}_S)$. In the absence of kinematical singularities, the algorithmic complexity also depends only linearly on the number of closure constraint equations, $\mathcal{N}_E$.

Reference [13] describes extensions to flexible multibody systems. The spatial operator formulation for flexible multibody systems is identical in form to the formulation for rigid multibody systems described in this paper. The structure of the dynamics algorithms requires the addition of only a few straightforward steps to handle body flexibility.

Acknowledgment

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. K. Kreutz-Delgado is also supported by an NSF PYI and a CalSpace grant.

Appendix A: Recursive Computational Algorithms

Based on the special structure of ϕ, ψ etc., it is possible to evaluate many of the dynamical expressions in a recursive manner and we describe some recursive algorithms in this appendix. First we define some notational shorthand to simplify the description of these algorithms:

$$(n_j + 1) \Rightarrow 1_{i(j)}$$

$$y(1_j, 0_j)x(0_j) \Rightarrow \sum_{m \in i^{-1}(j)} y(1_j, n_m)x(n_m)$$

$$y(1_j, 0_j)x(0_j)y^*(1_j, 0_j) \Rightarrow \sum_{m \in i^{-1}(j)} y(1_j, n_m)x(n_m)y^*(1_j, n_m) \tag{38}$$

where $y(\cdot, \cdot)$ and $x(\cdot)$ stand for some appropriate arrays. Thus where ever in an algorithm, a term with indices as in the left column appears, its meaning is actually given by the corresponding term in the column on the right.

- A recursive method for the computation of the block diagonal elements of P as defined by equation (25) and the entries of D, G, K, $\mathscr{E}_\psi$ and $\bar{\tau}$ defined in

equation (26) is given by:

$$
\begin{array}{l}
\text{for } j = 1 \cdots \ell \\
\quad \left\{ \begin{array}{l}
\qquad \text{If } i^{-1}(j) = \emptyset, \text{ then } P(0_j) = 0 \\
\text{for } k = 1_j \cdots n_j \\
\qquad P(k) = \psi(k, k-1)P(k-1)\psi^*(k, k-1) + M(k) \\
\qquad D(k) = H(k)P(k)H^*(k) \\
\qquad G(k) = P(k)H^*(k)D^{-1}(k) \\
\qquad \bar{\tau}(k) = I - G(k)H(k) \\
\quad \psi(k+1, k) = \phi(k+1, k)\bar{\tau}(k) \\
\quad K(k+1, k) = \phi(k+1, k)G(k) \\
\text{end loop}
\end{array} \right. \\
\text{end loop}
\end{array}
\tag{39}
$$

- The recursive computation of $\dot{B}^f = [I - H\psi K]^* D^{-1}[I - H\psi K]\bar{T}$ in equation (31) in STEP (A) can be carried out via the $O(N)$ tree topology forward dynamics algorithm described below. It also results in the computation of $\alpha^f = \psi^* D^{-1}[I - H\psi K]\bar{T}$ required in STEP (B) as an intermediate quantity.

$$
\begin{array}{l}
\left\{ \begin{array}{l}
\text{for } j = 1 \cdots \ell \\
\quad \left\{ \begin{array}{l}
\qquad \text{If } i^{-1}(j) = \emptyset, \text{ then } z(0_j) = 0, \bar{T}(0_j) = 0 \\
\text{for } k = 1_j \cdots n_j \\
\qquad z(k) = \psi(k, k-1)z(k-1) + K(k, k-1)\bar{T}(k-1) \\
\qquad \epsilon(k) = T(k) - H(k)z(k) \\
\qquad \nu(k) = D^{-1}(k)\epsilon(k) \\
\text{end loop}
\end{array} \right. \\
\text{end loop}
\end{array} \right. \\
\left\{ \begin{array}{l}
\qquad \alpha^f(n_\ell + 1) = 0 \\
\text{for } j = \ell \cdots 1 \\
\quad \left\{ \begin{array}{l}
\text{for } k = n_j \cdots 1_j \\
\qquad \alpha^f(k) = \psi^*(k+1, k)\alpha^f(k+1) + H^*(k)\nu(k) \\
\qquad \dot{\beta}^f(k) = \nu(k) - K^*(k+1, k)\alpha^f(k+1) \\
\text{end loop}
\end{array} \right. \\
\text{end loop}
\end{array} \right.
\end{array}
\tag{40}
$$

- STEP (C) requires the computation of $\Lambda = B^*\Omega B$. In order to obtain a $O(N)$ recursive scheme for the computation of Ω we first define the matrix Υ as the one satisfying the equation:

$$H^*D^{-1}H = \Upsilon - \mathcal{C}_\psi^*\Upsilon\mathcal{C}_\psi \tag{41}$$

Υ as defined above is a block diagonal matrix and its elements can be computed recursively. We now obtain the following decomposition of Ω.

Lemma 5:

$$\Omega = \Upsilon + \tilde{\psi}^*\Upsilon + \Upsilon\tilde{\psi} \tag{42}$$

Proof: See Appendix B. ∎

Noting that $\tilde{\psi}$ is strictly lower triangular, we can then recognize that Υ as nothing but the diagonal elements of Ω. We now present a recursive scheme to compute the block diagonal elements of Υ and of Ω.

$$\left\{\begin{array}{l} \Upsilon(n_\ell + 1) = 0 \\ \text{for } j = \ell \cdots 1 \\ \quad \left\{\begin{array}{l} \text{for } k = n_j \cdots 1_j \\ \quad \Upsilon(k) = \psi^*(k+1,k)\Upsilon(k+1)\psi(k+1,k) + H^*(k)D^{-1}(k)H(k) \\ \quad \left\{\begin{array}{l} \Omega(k,k) = \Upsilon(k) \\ \text{for } m = k-1 \cdots 1_j \\ \quad \Omega(k,m) = \Omega^*(m,k) = \Upsilon(k,m+1)\psi(m+1,m) \\ \text{end loop} \end{array}\right. \\ \text{end loop} \end{array}\right. \\ \text{end loop} \end{array}\right.$$

The above recursion yields the elements Ω_j on the block diagonal of Ω. Since Ω is symmetric, the off-diagonal elements satisfy $\Omega_{j,l} = \Omega_{l,j}^*$, and can be computed from the diagonal elements as follows. $\Omega_{l,j}$ for $l \in 1 \cdots (j-1)$ can be obtained via the following recursive scheme:

$$\left\{\begin{array}{l} \text{if } i^p(l) = j \text{ for some } p > 0 \\ \quad \left\{\begin{array}{l} \text{for } k = n_j \cdots 1_j \\ \quad \left\{\begin{array}{l} \text{for } m = n_l \cdots 1_l \\ \quad \Omega(k,m) = \Omega^*(m,k) = \Omega(k,1_j)\psi(1_j,m) \\ \text{end loop} \end{array}\right. \\ \text{end loop} \end{array}\right. \\ \text{else} \\ \quad \Omega_{j,l} = \Omega_{l,j}^* \equiv 0 \\ \text{end if} \end{array}\right. \tag{43}$$

- The $O(N)$ recursive implementation of $\beta^\delta = -[I - H\psi K]^* D^{-1} H\psi B A^* \hat{f}$ in equation (36) in STEP (E) is given below:

$$
\begin{cases}
\text{Define } \hat{x} \triangleq -BA^*\hat{f} \\
\text{for } j = 1 \cdots \ell \\
\quad \begin{cases}
\text{If } i^{-1}(j) = \varnothing, \text{ then } z(0_j) = 0,\ \hat{x}(0_j) = 0 \\
\text{for } k = 1_j \cdots n_j \\
\quad z(k) = \psi(k, k-1) z(k-1) + K(k, k-1)\hat{x}(k-1) \\
\quad \epsilon(k) = -H(k) z(k) \\
\quad \nu(k) = D^{-1}(k)\epsilon(k) \\
\text{end loop}
\end{cases} \\
\text{end loop}
\end{cases}
$$

$$
\begin{cases}
\alpha^\delta(n_\ell + 1) = 0 \\
\text{for } j = \ell \cdots 1 \\
\quad \begin{cases}
\text{for } k = n_j \cdots 1_j \\
\quad \alpha^\delta(k) = \psi^*(k+1, k)\alpha^\delta(k+1) + H^*(k)\nu(k) \\
\quad \dot{\beta}^\delta(k) = \nu(k) - K^*(k+1, k)\alpha^\delta(k+1) \\
\text{end loop}
\end{cases} \\
\text{end loop}
\end{cases}
\tag{44}
$$

Appendix B: Proofs of the Lemmas

In this appendix we give the proofs of the various lemmas. First we establish a few identities.

Lemma 6: P satisfies the equation

$$M = P - \mathcal{E}_\psi P \mathcal{E}_\psi^* \tag{45}$$

Proof: It is easy to verify from the definitions in equation (26) that

$$\bar{\tau} P \tau^* = 0 \quad \text{and thus} \quad \bar{\tau} P = \bar{\tau} P \bar{\tau}^*$$

Thus we can rewrite equation (25) in the form

$$M = P - \mathcal{E}_\phi [I - PH^* D^{-1} H] P \mathcal{E}_\phi^* = P - \mathcal{E}_\phi \bar{\tau} P \mathcal{E}_\phi^* = P - \mathcal{E}_\phi \bar{\tau} P \bar{\tau}^* \mathcal{E}_\phi$$
$$= P - \mathcal{E}_\psi P \mathcal{E}_\psi \quad \blacksquare$$

Lemma 7: In the stacked notation we have that:

(a) $$\tilde{\psi} \triangleq \psi \mathcal{E}_\psi = \mathcal{E}_\psi \psi = \psi - I \tag{46}$$

(b) $$\psi M \psi^* = P + \tilde{\psi} P + P \tilde{\psi}^* \tag{47}$$

(c)
$$\psi^{-1}\phi = I + KH\phi$$
$$\phi\psi^{-1} = I + \phi KH \tag{48}$$

(d)
$$H\psi M\psi^* H^* = D \tag{49}$$

Proof:

(a) From equation (27) we have that $\mathcal{E}_\psi = I - \psi^{-1}$. Substituting this into equation (46) gives the result.

(b) Pre- and post-multiplying equation (45) by ψ and ψ^* we have that
$$\psi M\psi^* = (\tilde{\psi} + I)P(\tilde{\psi} + I)^* - \tilde{\psi}P\tilde{\psi}^* = P + \tilde{\psi}P + P\tilde{\psi}^*$$

(c) We have from equation (26) and equation (27) that
$$\psi^{-1} = I - \mathcal{E}_\psi = I - \mathcal{E}_\phi\bar{\tau} = (I - \mathcal{E}_\phi) + \mathcal{E}_\phi\tau = \phi^{-1} + \mathcal{E}_\phi GH$$
$$= \phi^{-1} + KH \tag{50}$$
Pre- and post-multiplying this by ϕ gives the result.

(d) From equation (26)
$$\tau PH^* = PH^* \quad \text{so that} \quad \bar{\tau}PH^* = 0 \quad \text{and thus} \quad \mathcal{E}_\psi PH^* = 0$$
Pre- and post-multiplying equation (47) with H and H^* respectively and using the above fact in conjunction with equation (46) yields the result. ∎

Proof of Lemma 1:

We have that
$$\mathcal{M} = H\phi M\phi^* H^* = H(\phi\psi^{-1})\psi M\psi^*(\phi\psi^{-1})^* H^*$$
$$= H[I + \phi KH]\psi M\psi^*[I + \phi KH]^* H^* = [I + H\phi K]H\psi M\psi^* H^*[I + H\phi K]^*$$
$$= [I + H\phi K]D[I + H\phi K]^*$$
We have used equation (48) and equation (49) above. ∎

Proof of Lemma 2:

Using the operator identity in equation (50) along with a standard matrix identity, it follows that $(I + AB)^{-1} = I - A(I + BA)^{-1}B$, we have that
$$[I + H\phi K]^{-1} = I - H[I + \phi KH]^{-1}\phi K = I - H(\phi\psi^{-1})^{-1}\phi K = I - H\psi K$$
We have used equation (48) to simplify the above. ∎

Proof of Lemma 4:

We have that
$$[I - H\psi K]H\phi = H\phi - H\psi(KH\phi) = H\phi - H\psi(\psi^{-1}\phi - I) = H\psi$$
We have used equation (48) in the above for simplification. ∎

Proof of Lemma 5: Using equation (41) and equation (46) we have that
$$\Omega = \psi^* H^* D^{-1} H\psi = \psi^* \Upsilon\psi - \tilde{\psi}^* \Upsilon\tilde{\psi} = \Upsilon + \tilde{\psi}^* \Upsilon + \Upsilon\tilde{\psi} \quad ∎$$

References

[1] VERESHCHAGIN, A. F. "Computer Simulation of the Dynamics of Complicated Mechanisms of Robot-Manipulators," *Engineering Cybernetics,* Vol. 6, 1974, pp. 65–70.

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

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

[4] BRANDL, H., JOHANNI, R., and OTTER, M. "A Very Efficient Algorithm for the Simulation of Robots and similar Multibody Systems without Inversion of the Mass Matrix," IFAC/IFIP/IMACS Symposium (1st), Vienna, Austria, 1986, pp. 95–100.

[5] BAE, D. S., and HAUG, E. J. "A Recursive Formulation for Constrained Mechanical System Dynamics: Part I. Open loop systems," *Mechanics of Structures and Machines,* Vol. 15, No. 3, 1987, pp. 359–382.

[6] ROSENTHAL, D. E. "An Order n Formulation for Robotic Systems," *The Journal of the Astronautical Sciences,* Vol. 38, No. 4, 1990, pp. 511–530.

[7] RODRIGUEZ, G. "Kalman Filtering, Smoothing, and Topological Tree Dynamics," VPI/SU Symposium on Control of Large Structures, Blacksburg, Virginia, June 1987.

[8] RODRIGUEZ, G., and KREUTZ, K. "Recursive Mass Matrix Factorization and Inversion: An Operator Approach to Manipulator Forward Dynamics," *IEEE Transactions on Robotics and Automation,* 1990 (to appear); also, JPL Publication 88-11, 1988.

[9] RODRIGUEZ, G., KREUTZ, K., and JAIN, A. "A Spatial Operator Algebra for Manipulator Modeling and Control," *The International Journal of Robotics Research,* August 1991.

[10] BRANDL, H., JOHANNI, R., and OTTER, M. "An Algorithm for the Simulation of Multibody Systems with Kinematic Loops," World Congress on the Theory of Machines and Mechanisms (7th), Seville, Spain, 1987.

[11] BAE, D. S., and HAUG, E. J. "A Recursive Formulation for Constrained Mechanical System Dynamics: Part II. Closed loop systems," *Mechanics of Structures and Machines,* Vol 15, No. 4, 1987–88, pp. 481–506.

[12] BRYSON, A. E., Jr., and HO, Y. C. *Applied Optimal Control,* Blaisdell Publishing Company, Waltham, Massachusetts, 1969.

[13] JAIN, A., and RODRIGUEZ, G. "Recursive Dynamics for Flexible Multibody Systems using Spatial Operators," JPL Publication 90-26, Jet Propulsion Laboratory, Pasadena, California, December 1990.

[14] 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.

[15] KANE, T. R. and LEVINSON, D. *Dynamics: Theory and Applications,* McGraw-Hill, New York, 1985.

[16] RODRIGUEZ, G. "Random Field Estimation Approach to Robot Dynamics," *IEEE Transactions on Systems, Man, and Cybernetics,* September-October 1990, pp. 1081–1093.

[17] KAILATH, T. "The Innovations Approach to Detection and Estimation Theory," *Proceedings of the IEEE,* Vol. 58, March 1970, pp. 680–695.