

Published in final edited form as:

*Multibody Syst Dyn.* 2011 October 1; 26(3): 335–365. doi:10.1007/s11044-011-9267-6.

## Graph Theoretic Foundations of Multibody Dynamics Part II: Analysis and Algorithms

**Abhinandan Jain**

Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, California 91109

Abhinandan Jain: Abhi.Jain@jpl.nasa.gov

### Abstract

This second, of a two part paper, uses concepts from graph theory to obtain a deeper understanding of the mathematical foundations of multibody dynamics. The first part [7] established the *block-weighted adjacency (BWA)* matrix structure of spatial operators associated with serial and tree topology multibody system dynamics, and introduced the notions of *spatial kernel operators (SKO)* and *spatial propagation operators (SPO)*. This paper builds upon these connections to show that key analytical results and computational algorithms are a direct consequence of these structural properties and require minimal assumptions about the specific nature of the underlying multibody system. We formalize this notion by introducing the notion of *SKO models* for general tree-topology multibody systems. We show that key analytical results, including mass matrix factorization, inversion, and decomposition hold for all SKO models. It is also shown that key low-order scatter/gather recursive computational algorithms follow directly from these abstract-level analytical results. Application examples to illustrate the concrete application of these general results are provided. The paper also describes a general recipe for developing SKO models. The abstract nature of SKO models allows the application of these techniques to a very broad class of multibody systems.

### 1 Introduction

For tree-topology multibody systems, spatial operator techniques have been used to establish important analytical results, such as decompositions and factorizations of the mass matrix, and operator expressions for its inverse [9, 15, 16]. This analytical groundwork has led to low-order computational algorithms, such as for inverse dynamics, forward dynamics, and for computing the mass matrix. While spatial operators have been used to demonstrate common mathematical themes across a broad range of multibody systems, there has been a gap in our understanding of the underlying system properties that make such convergence possible.

To address this gap, part I of this paper [7] introduced the notions of *Spatial Kernel Operators (SKO)* and the related *Spatial Propagation Operators (SPO)*. It showed that key spatial operators are in fact instances of SKO and SPO matrices. The SKO and SPO matrices are themselves instances of *block-weighted adjacency (BWA)* and their 1-resolvent matrices for the graphs associated with tree-topology multibody systems. Thus it was seen that SKO and SPO properties of spatial operators applied very generally (serial/tree topology, rigid/flex bodies) with minimal assumptions on the lower level properties of the underlying system. This remarkable consistency persists despite significant differences in the component structure of the corresponding spatial operators for these different systems.

In this paper<sup>1</sup> we explore analytical results and computational algorithms that result from just the SKO and SPO properties of the spatial operators, and independent of the specific properties of the multibody system. This level of abstraction ensures that these results directly apply to a large family of multibody systems. We use examples and important robotics modeling and control applications to illustrate the connections between the abstract results and concrete dynamics problems. We introduce the notion of SKO models in Section 2 to formalize the specific dynamics models under consideration. In Section 3 we begin examining the properties of SKO and SPO operators at an abstract level, with minimal assumptions about the nature of the underlying multibody system other than that it has a tree-topology. This section introduces techniques for directly mapping SPO expressions into low-order, recursive scatter/gather algorithms when evaluation of operator expressions is needed. This process is illustrated for the classic inverse dynamics problem. Section 4 looks at the solutions of forward and backward Lyapunov equations associated with SKO models. It develops decompositions of SPO operator quadratic products. These results are used to study the structure of mass matrices as well to develop algorithms for efficiently computing them. Section 5 focuses on solving the quadratic Riccati equation for SKO models. The solution provides a generalization of the family of articulated body inertia quantities. Several SKO and SPO identities are developed in this section, that are used in Section 6 to develop general expressions for the factorization and inversion of the mass matrix and the operational space inertia. These results are used to develop the general form of the classic  $O(\mathcal{N})$  AB algorithm for the forward dynamics problem. Section 7 examines the system topology dependent sparsity structure of the spatial operators and the mass matrix. Section 8 describes the process for developing SKO models for general multibody systems.

## 2 SKO models

Central to SKO formulations are the development of SKO models for multibody systems. These models are formally defined in the following section, and their properties are explored in the rest of the paper.

### 2.1 Definition of SKO models

An **SKO model** for an  $n$ -links tree-topology multibody system consists of the following:

1. A tree digraph reflecting the bodies and their connectivity in the system.
2. An  $\varepsilon_A$  SKO operator and associated SPO operator,  $\mathbb{A} \triangleq (\mathbf{I} - \varepsilon_A)^{-1}$ .
3. A full-rank block-diagonal, joint map matrix operator,  $H$ .
4. A block-diagonal and positive-definite spatial inertia operator,  $\mathbf{M}$ .
5. Stacked vectors:  $\dot{\theta}$  denoting independent generalized velocities,  $\mathcal{T}$  the generalized forces,  $\mathcal{V}$  the node velocities,  $\alpha$  the node accelerations,  $\mathbf{f}$  the inter-node forces,  $\mathbf{a}$  the node Coriolis accelerations,  $\mathbf{b}$  the node gyroscopic forces the system,  $\mathcal{N}$  the number of degrees of freedom, and the equations of motion defined as:

$$\begin{aligned}\mathcal{V} &= \mathbb{A}^* H^* \dot{\theta} \\ \alpha &= \mathbb{A}^* (H^* \ddot{\theta} + \mathbf{a}) \\ \mathbf{f} &= \mathbb{A}(\mathbf{M}\alpha + \mathbf{b}) \\ \mathcal{T} &= H\mathbf{f}\end{aligned}$$

<sup>1</sup>Preliminary results in this paper have been reported in [8].

(1)

Thus,

$$\mathcal{T} = \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta, \dot{\theta})$$

(2)

where

$$\mathcal{M}(\theta) \triangleq H \mathbb{A} M \mathbb{A}^* H^* \in \mathcal{R}^{\mathcal{N} \times \mathcal{N}} \text{ and } \mathcal{C}(\theta, \dot{\theta}) \triangleq H \mathbb{A} (M \mathbb{A}^* \mathbf{a} + \mathbf{b}) \in \mathcal{R}^{\mathcal{N}}$$

(3)

$\mathcal{M}$  is the symmetric and positive-definite mass matrix for the tree system and  $\mathcal{C}$  is the vector of nonlinear Coriolis and gyroscopic velocity dependent terms. Eq. 3 is the *Newton-Euler operator factorization* of the mass matrix.

To maximize generality, no assumptions have been made about the specific nature of the weight matrices in the SKO operators, about the component elements of  $H$  and  $M$ , or that the tree digraph is the *standard digraph*<sup>2</sup> in an SKO model.

## 2.2 Existence of SKO models

The dynamics formulations derived in [7] for rigid-body, serial-chain and tree-topology, multibody systems are examples of SKO models with  $\varepsilon_{\mathbb{A}}$  and  $\varphi$  as the respective  $\varepsilon_{\mathbb{A}}$  SKO and  $\mathbb{A}$  SPO operators. The existence of an SKO model does not depend on any exceptional or unusual features of tree-topology multibody systems. Indeed, SKO models can be viewed as a refinement of the models obtained from the classic, and widely used, *Kane's method* for developing multibody dynamics models [10, 11]. Central to Kane's method is the identification of *partial velocities* that define the mapping from generalized velocities to body velocities. The elements of the  $\mathbb{A}^* H^*$  matrix product of an SKO model are precisely Kane's partial velocities. This is an important connection between these modeling techniques.

While the partial velocity elements are adequate for formulating dynamics models using Kane's method, the important SPO property is unfortunately lost once the  $\mathbb{A}^* H^*$  partial velocities product is formed. The SPO nature of the operator  $\mathbb{A}$  plays a crucial role in analysis and algorithm development. Therefore, SKO models retain the more fundamental  $(H, \mathbb{A}, M)$  family of operators to not only define the Eq. 2 dynamics model, but also to derive important analytical and algorithmic techniques. We will, at times refer to an SKO model by its  $(H, \mathbb{A}, M)$  triplet of operators. Section 8 describes a systematic process for developing SKO models for multibody systems.

## 2.3 Generalizations of SKO models

To highlight the possible areas of generalization of SKO models, we contrast, below, the properties of BWA matrices with those of the familiar SKO operators for serial and tree-topology rigid body systems.

- BWA weight matrices are not limited to  $6 \times 6$   $\varphi(\mathcal{P}(k), k)$  rigid-body transformation matrices.

<sup>2</sup>The **standard** digraph associated with a multibody system has the inertial frame as the root node, and all the links in the system as the remaining nodes in the digraph [7].

- BWA weight matrices can be non-square.
- BWA weight matrices can be of non-uniform size across edges.
- BWA weight matrices can be singular.

In general, SKO weight matrices can, and will, deviate in all of these respects from those for rigid body systems. We will see that the wide variability of the multibody systems is embodied in the definitions of their SKO model digraphs and, in the specific definitions of the weight matrices and operator component values.

We now mention two cautionary points concerning SKO models.

**Definition of body nodes:** The *standard* digraph for a multibody system assigns a node to each physical body in the system. This one-to-one correspondence between physical bodies and digraph nodes has been used to develop the SKO models for tree-topology systems. However, this one-to-one correspondence is not a requirement for an SKO model. We will encounter cases where multiple physical bodies are associated with a single node in the digraph, as well as the converse, where multiple nodes are assigned to a single body! Since the majority of models will indeed be using the standard digraphs, we will use the *nodes* and *bodies* terminology interchangeably, and will highlight the exceptional cases when we encounter them.

**Spatial operator compatibility:** By construction, spatial operators have block structure derived from a combination of the underlying model digraph and the node weight dimensions. However, SKO models for a system are not unique, and may differ in the choices of digraph nodes, edges and weight dimensions.

Spatial operators and stacked vectors are said to be **compatible** when they are associated with the same SKO model. Expressions composing spatial operators and stacked vectors are only meaningful when all the component terms are *compatible*. Therefore, care is needed to avoid mixing spatial operators associated with different SKO models. The compatibility requirement, however, does not preclude the definition of transformations that relate operators across different SKO models.

In this section, and the following ones, we develop analytical results and algorithms for SKO models with no assumptions beyond those listed for the definition of these models. Therefore, the results can be used for any system for which an SKO model is available. We will also apply these results to important dynamics problems. Several of the spatial operator techniques and algorithms for serial-chain, rigid body systems will be generalized to the broader class of SKO models.

### 3 SPO operator/vector products for trees

In addition to the mathematical analysis supported by spatial operators, there is an intimate relationship between operator expressions and efficient recursive computational algorithms. We begin by studying this relationship in detail in the following lemma.

**Lemma 1 Tips-to-base gather recursion to evaluate  $\mathbb{A} \mathbf{x}$** —Given an SPO operator,  $\mathbb{A}$ , and a compatible stacked vector,  $\mathbf{x}$ , for a tree-topology system, the  $y(k)$  elements of  $\mathbf{y} = \mathbb{A} \mathbf{x}$ , satisfy the following parent/child recursive relationship:

$$y(k) = \sum_{Vi \in C(k)} \mathbb{A}(k, i) y(i) + x(k) \quad (4)$$

Based on this relationship, the elements of  $y$  can be computed using the following  $O(\mathcal{N})$  **tips-to-base gather** recursive procedure:

$$\left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (tips - to - base gather)} \\ y(k) = \sum_{\forall i \in C(k)} \mathbb{A}(k, i) y(i) + x(k) \\ \text{end loop} \end{array} \right. \quad (5)$$

As will be customary, the steps in such gather recursions only apply to the body nodes and not the inertial node in the digraph.

**Proof:** The  $y = \mathbb{A}x$  expression implies that

$$y(k) = \sum_{j=1}^n \mathbb{A}(k, j) x(j) = \sum_{\forall j \leq k} \mathbb{A}(k, j) x(j) \quad (6)$$

Therefore,

$$y(k) \stackrel{6}{=} \sum_{\forall i \in C(k)} \mathbb{A}(k, i) \sum_{\forall j \leq i} \mathbb{A}(i, j) x(j) + x(k) \stackrel{6}{=} \sum_{\forall i \in C(k)} \mathbb{A}(k, i) y(i) + x(k)$$

This establishes Eq. 4. The procedure in Eq. 5 follows directly from this recursive expression.

The left side of Figure 1 illustrates the flow of the recursive procedure in Eq. 5 for computing the  $y(k)$  elements. It starts at the tip bodies of the tree and proceeds towards the base-body, while gathering together values converging at the branches. The computational cost of this tips-to-base gather procedure is just of  $O(\mathcal{N})$  complexity, instead of the  $O(\mathcal{N}^2)$  complexity of a direct  $\mathbb{A}x$  matrix/vector product. We will encounter several applications later that use this lemma to efficiently evaluate operator expressions involving products of SPO operators and stacked vectors.

Similarly, the  $x = \mathbb{A}^*y$  product can be evaluated using a base-to-tips,  $O(\mathcal{N})$  scatter computational algorithm described in the following lemma.

**Lemma 2 Base-to-tips scatter recursion to evaluate  $\mathbb{A}^*x$** —Given an SPO operator,  $\mathbb{A}$ , and a compatible stacked vector  $x$ , for a tree-topology system, the  $y(k)$  elements of  $y = \mathbb{A}^*x$ , satisfy the following parent/child recursive relationship:

$$y(k) = \mathbb{A}^*(\varphi(k), k) y(\varphi(k)) + x(k) \quad (7)$$

Based on this relationship, the elements of  $y$  can be computed using the following  $O(\mathcal{N})$  **base-to-tips scatter** recursive procedure:

$$\left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (base - to - tips scatter)} \\ y(k) = \mathbb{A}^*(\varphi(k), k) y(\varphi(k)) + x(k) \\ \text{end loop} \end{array} \right. \quad (8)$$

As will be customary, the steps in such scatter recursions only apply to the body nodes and not the inertial nodes in the digraph. The quantities for the inertial node required in the initial step for the base-bodies in the scatter recursion are always assumed to be zero.

**Proof:** The  $y = \mathbb{A}^*x$  expression implies that

$$y(k) = \sum_{j=1}^n \mathbb{A}^*(j, k)x(j) = \sum_{\forall j \succ k} \mathbb{A}^*(j, k)x(j) \quad (9)$$

Therefore,

$$\begin{aligned} y(k) &\stackrel{9}{=} \mathbb{A}^*(\varphi(k), k) \sum_{\forall j \succ \varphi(k)} \mathbb{A}^*(i, \varphi(k))x(i) + x(k) \\ &\stackrel{9}{=} \mathbb{A}^*(\varphi(k), k)y(\varphi(k)) + x(k) \end{aligned}$$

This establishes Eq. 7. The recursion in Eq. 8 is a direct consequence of this relationship. The right side of Figure 1 illustrates the flow of the recursive procedure in Eq. 8 for computing the  $y(k)$  elements. It starts at the base-body of the tree and proceeds towards the tips, while scattering the values onto diverging branches. The computational cost of this procedure is again just of  $O(N)$  complexity, instead of the  $O(N^2)$  complexity of a direct  $\mathbb{A}^*x$  matrix/vector product. We will encounter several applications that use this lemma to efficiently evaluate products involving the transpose of an SPO operator and a stacked vector.

**Corollary 3.1 Recursive evaluation of  $\bar{\mathbb{A}}x$  and  $\bar{\mathbb{A}}^*x$** —Similar to Eq. 5 and Eq. 8, show that the elements of  $y = \bar{\mathbb{A}}x$  satisfy the following recursive gather relationship:

$$y(k) = \sum_{\forall j \in C(k)} \bar{\mathbb{A}}(k, j) [y(j) + x(j)]$$

Additionally, show that the elements of  $y = \bar{\mathbb{A}}^*x$  satisfy the following recursive scatter relationship:

$$y(k) = \bar{\mathbb{A}}^*(\varphi(k), k) [y(\varphi(k)) + x(\varphi(k))]$$

**Proof:** We have that  $y \triangleq \bar{\mathbb{A}}x = \bar{\mathbb{A}}\bar{y} - x$  where  $\bar{y} = \bar{\mathbb{A}}x$ . This implies that

$$y(k) = \bar{y}(k) - x(k) \stackrel{5}{=} \sum_{\forall i \in C(k)} \bar{\mathbb{A}}(k, i)\bar{y}(i) = \sum_{\forall i \in C(k)} \bar{\mathbb{A}}(k, i)[y(i) + x(i)]$$

Thus, the computational algorithm for  $y$  is the one in Eq. 5 except that the computational step in the loop is now given by the above expression with initial condition  $y(0) = 0$ .

Similarly  $y \triangleq \bar{\mathbb{A}}^*x = \bar{y} - x$  where  $\bar{y} = \bar{\mathbb{A}}^*x$ . It thus follows that the computational algorithm for  $\bar{\mathbb{A}}^*x$  is the same as in Eq. 8 except that the computational step in the loop is now given by

$$y(k) = \mathbb{A}^*(\varphi(k), k)[y(\varphi(k)) + x(\varphi k)]$$

with initial condition  $y(n+1) = 0$ .

Based on the results in this section, we can make the important observation that the serial-chain recursive algorithms for operator expressions can be easily generalized to tree-topology systems by simply replacing tip-to-base recursions with tips-to-base gather recursions, and base-to-tip recursions with base-to-tips scatter recursions.

### 3.1 SKO model $O(\mathcal{N})$ Newton-Euler inverse dynamics

We now apply the mapping of SPO and stacked vector products to develop an efficient  $O(\mathcal{N})$  recursive algorithm for the inverse dynamics problem for a system with an SKO model. For the inverse dynamics problem, the  $\bar{\theta}$  generalized accelerations are assumed to be known, and the corresponding  $\mathcal{T}$  generalized forces need to be computed. Eq. 1 and Eq. 2 define the operator expressions for this mapping for the system. These operator expressions are precisely in the form of SPO and stacked vector products. Thus Lemmas 1 and 2 can be used to evaluate them using recursive scatter and gather algorithms. Algorithm 3.1 describes such an implementation. The algorithm begins with a base-to-tips scatter recursion that implement the operator expressions for  $\nu$  and  $\alpha$  in Eq. 1. This is followed by a recursive tips-to-base gather recursion to evaluate the expressions for  $f$  and  $\mathcal{T}$  in Eq. 1.

#### Algorithm 3.1 Newton-Euler inverse dynamics algorithm for an SKO model

$$\left\{ \begin{array}{l} \text{for all nodes } k \text{ (base-to-tips scatter)} \\ \quad \nu(k) = \mathbb{A}^*(\varphi(k), k)\nu(\varphi(k)) + H^*(k)\dot{\bar{\theta}}(k) \\ \quad \alpha(k) = \mathbb{A}^*(\varphi(k), k)\alpha(\varphi(k)) + H^*(k)\ddot{\bar{\theta}}(k) + a(k) \\ \text{end loop} \\ \\ \left\{ \begin{array}{l} \text{for all nodes } k \text{ (tips-to-base gather)} \\ \quad f(k) = \sum_{\forall j \in \mathcal{C}(k)} \mathbb{A}(k, j)f(j) + M(k)\alpha(k) + b(k) \\ \quad \mathcal{T}(k) = H(k)f(k) \\ \text{end loop} \end{array} \right. \end{array} \right.$$

(10)

## 4 Lyapunov equations for SKO models

This section studies SPO operator quadratic expressions that are related to forward and backward Lyapunov equations for an SKO model.

### 4.1 Forward Lyapunov recursions for SKO models

**Lemma 3 Structure of  $\mathbb{A}X\mathbb{B}^*$** —Let  $\mathbb{A}$  and  $\mathbb{B}$  denote compatible SPO operators and  $X$  be a compatible block-diagonal operator. Then the  $Z \triangleq \mathbb{A}X\mathbb{B}^*$  product can be decomposed into the following disjoint sum

$$Z=Y+\tilde{A}Y+Y\tilde{B}^* \quad (11)$$

where  $Y$  is a block-diagonal operator satisfying the **forward Lyapunov equation**:

$$X=Y - \mathcal{E}_A Y \mathcal{E}_B^* \quad (12)$$

The  $Y(k)$  diagonal elements satisfy the recursive parent/child relationship:

$$Y(k) = \sum_{\forall j \in \mathcal{C}(k)} A(k, j) Y(j) B^*(k, j) + X(k) \quad (13)$$

Based on this relationship, the  $Y(k)$  terms can be computed via the  $O(\mathcal{N})$  tips-to-base, forward Lyapunov gather recursion:

$$\left\{ \begin{array}{l} \text{for all nodes } k \text{ (tips - to - base gather)} \\ y(k) = \sum_{\forall j \in \mathcal{C}(k)} A(k, j) Y(j) B^*(k, j) + X(k) \\ \text{end loop} \end{array} \right. \quad (14)$$

While  $Y$  defines the block-diagonal elements of  $Z$ , the following recursive expressions describe all the  $Z(i, j)$  terms, including the off-diagonal ones:

$$Z(i, j) = \begin{cases} Y(i) & \text{for } i=j \\ A(i, k) Z(k, j) & \text{for } i > k \geq j, \quad k \in \mathcal{C}(i) \\ Z(i, k) B^*(j, k) & \text{for } i \leq k < j, \quad k \in \mathcal{C}(j) \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (15)$$

**Proof:** First let us verify that Eq. 12 is well-posed, i.e., that  $\mathcal{E}_A Y \mathcal{E}_B^*$  is block-diagonal when  $Y$  is block diagonal. This follows from:

$$\begin{aligned} \mathcal{E}_A Y \mathcal{E}_B^* &= \left\{ \sum_{j=1}^n e_{\varphi(j)} A(\varphi(j), j) e_j^* \right\} Y \left\{ \sum_{k=1}^n e_k B^*(\varphi(k), k) e_{\varphi(k)}^* \right\} \\ &= \sum_{j=1}^n e_{\varphi(j)} A(\varphi(j), j) Y(j) B^*(\varphi(j), j) e_{\varphi(j)}^* \end{aligned} \quad (16)$$

This has the form of a block-diagonal matrix with the  $k^{\text{th}}$  diagonal element being  $\sum_{j \in \mathcal{C}(k)} A(k, j) Y(j) B^*(k, j)$ . This implies that, at the component level, Eq. 12 is equivalent to

$$X(k) \stackrel{12,16}{=} Y(k) - \sum_{\forall j \in \mathcal{C}(k)} A(k, j) Y(j) B^*(k, j)$$

from which Eq. 13 follows.

Pre and post multiply Eq. 12 by  $A$  and  $B^*$  to get:

$$A X B^* = (\tilde{A} + I) Y (\tilde{B}^* + I) - \tilde{A} Y \tilde{B}^* = \tilde{A} Y + Y \tilde{B}^* + Y$$



This establishes Eq. 11.

The middle pair of expressions in Eq. 15 follow directly from the structure of  $\tilde{A}Y$  and  $Y\tilde{B}^*$ .

The disjoint terms in Eq. 11 represent the following:

- The non-zero block-elements of the block-diagonal  $Y$  are the block-diagonal elements of  $\tilde{A}X\tilde{B}^*$ .
- The non-zero block-elements of  $\tilde{A}Y$  and  $Y\tilde{B}^*$  correspond to *related* node pairs in the system, i.e., nodes connected by a directed path.
- The remaining terms are zero, and correspond to *unrelated* node pairs, i.e., nodes with no directed path connecting them.

Since entries corresponding to node pairs that do not have an ancestor/child relationship are zero,  $\tilde{A}X\tilde{B}^*$  is in general sparse. The sparsity structure reflects the topological connectivity of the nodes. Based on Eq. 15, the strategy for computing  $\tilde{A}X\tilde{B}^*$  consists of:

1. First, compute the non-zero block-diagonal terms in  $Y$  using the tips-to-base gather algorithm in Eq. 14.
2. Now consider the case of computing  $Z(j, k)$ , where  $j$  is an ancestor of  $k$ . Assume that node  $i$  is the child of  $j$  on the path to  $k$ . Use Eq. 15 to compute  $Z(j, k) = \tilde{A}(j, i)Z(i, k)$ . This expression implies a recursion that starts with the  $Z(k, k) = Y(k, k)$  diagonal element, and computes the  $Z(i, k)$  elements recursively along the path to node  $j$ .
3. Now consider the converse case of computing  $Z(j, k)$ , where  $j$  is a descendant of  $k$ . Assume that node  $i$  is the child of  $k$  on the path to  $j$ . Use Eq. 15 to compute  $Z(j, k) = Z(j, i)\tilde{B}^*(k, i)$ . This expression implies a recursion that starts with the  $Z(j, j) = Y(j, j)$  diagonal element, and computes the  $Z(j, i)$  elements recursively along the path to node  $k$ .

In the special case where  $\tilde{A} = \tilde{B}$ , case (3) above is unneeded because  $Z$  is symmetric, and  $Z(j, k) = Z^*(k, j)$ .

## 4.2 Mass matrix computation for an SKO model

As seen in Eq. 2, the mass matrix in the SKO model has the operator expression  $\mathcal{M} = H\tilde{A}M\tilde{A}^*H^*$ . The following lemma uses Lemma 3 to derive a decomposition of, and computational procedure for, the mass matrix.

**Lemma 4 SKO model mass matrix decomposition**—The SKO model mass matrix can be decomposed into a sum of disjoint terms as follows:

$$M = HRH^* + H\tilde{A}\tilde{R}H^* + HR\tilde{A}^*H^* \quad (17)$$

where the block-diagonal composite body inertia operator,  $\tilde{R}$ , satisfies the following forward Lyapunov equation:

$$M = \tilde{R} - \mathcal{E}_{\tilde{A}}\tilde{R}\mathcal{E}_{\tilde{A}}^* \quad (18)$$

**Proof:** The results here are a direct consequence of Lemma 3, once we identify  $\tilde{B} = \tilde{A}$ , and  $X = M$ . From the lemma, we have the following disjoint decomposition of  $\tilde{A}M\tilde{A}^*$ :

$$\mathbb{A}M\mathbb{A}^* = \mathcal{R} + \tilde{\mathbb{A}}\mathcal{R} + \mathcal{R}\tilde{\mathbb{A}}^* \quad (19)$$

Eq. 17 follows by pre and post multiplying the above expression with the block-diagonal  $H$  and  $H^*$ , respectively.

The tips-to-base gather Algorithm 4.1 describes the general composite body inertia algorithm for an SKO model.

**Algorithm 4.1 Recursive computation of composite body inertias for an SKO model**

$$\left\{ \begin{array}{l} \text{for all nodes } k \text{ (tips - to - base gather)} \\ \mathcal{R}(k) = \sum_{i \in C(k)} \mathbb{A}(k, i)\mathcal{R}(i)\mathbb{A}^*(k, i) + M(k) \\ \text{end loop} \end{array} \right. \quad (20)$$

Remark 5.2 in Jain [7] discussed the sparsity structure of the  $\mathcal{E}_{\mathbb{A}}$  and  $\mathbb{A}$  matrices for the canonical tree in Figure 2 in the paper. Based on Eq. 17, the following matrix illustrates the sparsity structure of the mass matrix for the same system:

$$\mathcal{M} = \begin{pmatrix} X & \cdot & X & \cdot & \cdot & X & X \\ \cdot & X & \cdot & X & \cdot & X & X \\ X & \cdot & X & \cdot & \cdot & X & X \\ \cdot & X & \cdot & X & \cdot & X & X \\ \cdot & \cdot & \cdot & \cdot & X & X & X \\ X & X & X & X & X & X & X \\ X & X & X & X & X & X & X \end{pmatrix}$$

The above matrix shows that, for trees, the  $\mathcal{M}$  mass matrix contains block-elements that are zero, because, unlike serial-chains, trees can have *unrelated* nodes. For serial-chain systems, the mass matrix is dense and fully populated because all nodes in a serial-chain system are *related*.

The sparsity of the mass matrix depends on the underlying branching structure of the system. As discussed in Section 7, the sparsity structure of the mass matrix is determined by the connectivity of its serial-chain segments; blocks corresponding to the segments are dense, while those corresponding to unrelated segments are zero. Section 7 also examines the sparsity structure of the mass matrix for the more complex tree system in Figure 3.

Based on the decomposition in Eq. 17, Algorithm 4.2 describes a recursive procedure for computing the mass matrix of a tree-topology system. The outer loop includes the tips-to-base gather computation of the composite body inertias. The inner loop computes the mass matrix cross-terms for all the ancestors of a body.

**Algorithm 4.2 Recursive computation of the SKO model mass matrix—**

$$\left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (tips-to-base gather)} \\ \mathcal{R}(k) = \sum_{\forall i \in \mathcal{G}(k)} \mathbb{A}(k, i) \mathcal{R}(i) \mathbb{A}^*(k, i) + M(k) \\ \left\{ \begin{array}{l} j = k, \quad X(k) = \mathcal{R}(k) H^*(k), \quad \mathcal{M}(k, k) = H(k) X(k) \\ \text{while } j \\ \quad l = \varphi(j) \\ \quad X(l) = \phi(l, j) X(j) \\ \quad \mathcal{M}(l, k) = \mathcal{M}^*(k, l) = H(l) X(l) \\ \quad j = l \\ \text{end loop} \end{array} \right. \\ \text{end loop} \end{array} \right.$$

### 4.3 Backward Lyapunov recursions for SKO models

Complementing the discussion on forward Lyapunov equations is the following lemma regarding backward Lyapunov equations for SKO models.

**Lemma 5 Structure of  $\mathbb{A}^* X \mathbb{B}$** —Let  $\mathbb{A}$  and  $\mathbb{B}$  denote compatible SPO operators and  $X$  be a compatible block-diagonal operator. Then the  $Z = \mathbb{A}^* X \mathbb{B}$  product can be expressed as the following sum of disjoint terms:

$$\text{where } R \triangleq \begin{array}{l} Z = Y + \tilde{\mathbb{A}}^* Y + Y \tilde{\mathbb{B}} + R \\ \sum_{\substack{\forall i, j : i \neq j \\ k = \varphi(i, j)}} e_i \mathbb{A}^*(k, i) Y(k) \mathbb{B}(k, j) e_j^* \end{array} \tag{21}$$

$Y$  is a block-diagonal operator satisfying the following backward Lyapunov equation:

$$X = Y - \text{diagOf} \{ \mathcal{E}_A^* Y \mathcal{E}_B \} \tag{22}$$

The  $\text{diagOf} \{ \mathcal{E}_A^* Y \mathcal{E}_B \}$  term represents just the diagonal block elements of the (generally non block-diagonal)  $\mathcal{E}_A^* Y \mathcal{E}_B$  matrix. The  $Y(k)$  diagonal elements satisfy the following parent/child recursive relationship:

$$Y(k) = \mathbb{A}^*(\varphi(k), k) Y(\varphi(k)) \mathbb{B}(\varphi(k), k) + X(k) \tag{23}$$

Based on this relationship, the  $Y(k)$  diagonal elements can be computed via the following  $O(N)$  base-to-tips scatter recursion:

$$\left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (base - to - tips scatter)} \\ y(k) = \mathbb{A}^*(\varphi(k), k) Y(\varphi(k)) \mathbb{B}(\varphi(k), k) + X(k) \\ \text{end loop} \end{array} \right. \tag{24}$$

While  $Y$  defines the block-diagonal elements of  $Z$ , the following recursive expressions describe all the  $Z(i, j)$  terms, including the off-diagonal ones:

$$Z(i, j) = \begin{cases} Y(i) & \text{for } i=j \\ \mathbb{A}^*(k, i) Z(k, j) & \text{for } i < k \leq j, \quad k = \varphi(i) \\ Z(i, k) \mathbb{B}(k, j) & \text{for } i \geq k > j, \quad k = \varphi(j) \\ \mathbb{A}^*(k, i) Y(k) \mathbb{B}(k, j) & \text{for } i \neq j, j \neq i, \quad k = \varphi(i, j) \end{cases} \tag{25}$$

**Proof:** We have

$$\begin{aligned} \mathcal{E}_A^* Y \mathcal{E}_B & \left\{ \sum_{j=1}^n e_j \mathbb{A}^*(\varphi(j), j) e_{\varphi(j)}^* \right\} Y \left\{ \sum_{k=1}^n e_{\varphi(k)} \mathbb{B}(\varphi(k), k) e_k^* \right\} \\ & = \sum_{\forall j,k: \varphi(j)=\varphi(k)} e_j \mathbb{A}^*(\varphi(j), j) Y(\varphi(j)) \mathbb{B}(\varphi(k), k) e_k^* \end{aligned}$$

Therefore,

$$\text{diagOf} \left\{ \mathcal{E}_A^* Y \mathcal{E}_B \right\} = \sum_{j=1}^n e_j \mathbb{A}^*(\varphi(j), j) Y(\varphi(j)) \mathbb{B}(\varphi(j), j) e_j^* \quad (26)$$

This implies that, at the component level, Eq. 22 is equivalent to

$$X(k) \stackrel{22,26}{=} Y(k) - \mathbb{A}^*(\varphi(j), j) Y(\varphi(j)) \mathbb{B}(\varphi(j), j)$$

from which Eq. 23 follows.

Now let us examine the elements of  $Z = \mathbb{A}^* X \mathbb{B}$  in Eq. 21.

$$\begin{aligned} Z & = \left\{ \sum_{i=1}^n \sum_{\forall m \geq i} e_i \mathbb{A}^*(m, i) e_m^* \right\} X \left\{ \sum_{j=1}^n \sum_{\forall l \geq j} e_l \mathbb{B}(l, j) e_j^* \right\} \\ & = \sum_{i=1}^n \sum_{\forall m \geq i, j} e_i \mathbb{A}^*(m, i) X(m) \mathbb{B}(m, j) e_j^* \end{aligned}$$

Hence,

$$Z(i, j) = \sum_{\forall m \geq i, j} \mathbb{A}^*(m, i) X(m) \mathbb{B}(m, j)$$

The expressions in Eq. 25 are a direct consequence of the above expression for  $Z(i, j)$ .

Unlike  $\mathbb{A} X \mathbb{B}^*$ ,  $\mathbb{A}^* X \mathbb{B}$  is a fully populated matrix. The disjoint partitioned terms in Eq. 21 have the following properties:

- The block-diagonal  $Y$  contains the diagonal block elements terms of  $\mathbb{A}^* X \mathbb{B}$ .
- The non-zero block-elements of  $\mathbb{A}^* Y$  and  $Y \mathbb{B}$  correspond to all *related* nodes.
- The non-zero block-elements of  $R$  on the other hand are for the remaining *unrelated* node pairs, i.e., ones that have no directed path connecting them.

As is evident from Eq. 25, all the block-elements of  $\mathbb{A}^* X \mathbb{B}$  intimately dependent on the elements of the  $Y$  block-diagonal matrix. An important application of this lemma is to operational space inertias, as discussed in Section 6.2.

The recursive computational strategy for the elements of  $\mathbb{A}^* X \mathbb{B}$  is described in Algorithm 4.3. For the special case where  $\mathbb{A} = \mathbb{B}$ ,  $Z$  is symmetric, and Step (3) in the algorithm can be

skipped since these elements can be obtained from the transposes of the corresponding elements in Step (2).

**Algorithm 4.3 Computation of  $Z = \mathbf{A}^* \mathbf{X} \mathbf{B}$** —We have four situations to consider when computing a  $Z(k, j)$  element. The respective recursive steps are described below and illustrated in Figure 2.

1. To obtain a block-diagonal element,  $Z(k, k)$ , compute the  $Y(k)$  element using the base-to-tips scatter  $O(\mathcal{N})$  algorithm described in Eq. 24 as illustrated in the diagram on the left in Figure 2.
2. If the  $k^{\text{th}}$  body is an ancestor of the  $j^{\text{th}}$  body,  $Z(j, k)$  can be computed recursively, starting with the  $Z(k, k) = Y(k)$  diagonal element, which, in turn, is computed using the process described in case (1). The recursion for  $Z(j, k)$ , illustrated by the middle diagram of Figure 2, starts with this  $Y(k)$  diagonal entry, and propagates it along the path from the  $k^{\text{th}}$  to the  $j^{\text{th}}$  body, using the second expression in Eq. 25 to compute the  $Z(k-1, k)$ ,  $Z(k-2, k)$ , etc., terms.
3. If, on the other hand, the  $k^{\text{th}}$  body is an ancestor of the  $j^{\text{th}}$  body, then  $Z(k, j)$  can be computed recursively, starting with the  $Z(k, k) = Y(k)$  diagonal element, which, in turn, is computed using the process described in case (1). The recursion for  $Z(k, j)$ , illustrated in the middle diagram of Figure 2, starts with this  $Y(k)$  diagonal entry, and propagates it along the path from the  $k^{\text{th}}$  to the  $j^{\text{th}}$  body, using the third expression in Eq. 25 to compute the  $Z(k, k-1)$ ,  $Z(k, k-2)$ , etc., terms.
4. Now consider the remaining case where the  $(k, j)$  node pair are unrelated. In this case, identify the body that is the closest ancestor for this pair of bodies. If the bodies do not have a common ancestor, then  $Z(k, j)$  is zero since the bodies belong to independent, decoupled multibody trees.

If, on the other hand, there is a common ancestor, denoted the  $i^{\text{th}}$  body, then the first step is to compute  $Z(k, i)$ , using the recursive procedure described in case (2). This value is then recursively propagated along the path from the  $i^{\text{th}}$  body to the  $j^{\text{th}}$  body, using the procedure in case (3) to compute the  $Z(k, i-1)$ ,  $Z(k, i-2)$ , etc., terms. This process is illustrated in the diagram on the right in Figure 2.

Later, Section 6.2 describes the use of these backwards Lyapunov equation results and algorithms for the operational space inertia matrices associated with tree multibody systems. The following Lemma describes simplifications of the backwards Lyapunov equation for serial-chain systems.

**Lemma 6  $\mathbf{A}^* \mathbf{X} \mathbf{B}$  structure for a serial-chain SKO model**—For a serial-chain system,  $Z = \mathbf{A}^* \mathbf{X} \mathbf{B}$  can be decomposed into disjoint diagonal, strictly upper triangular, and strictly lower triangular terms as follows:

$$Z = Y + \tilde{\mathbf{A}}^* Y + Y \tilde{\mathbf{B}} \quad (27)$$

where  $Y \in \mathbb{R}^{6n \times 6n}$  is a block-diagonal operator satisfying the following backward Lyapunov equation:

$$X = Y - \mathcal{E}_A^* Y \mathcal{E}_B \quad (28)$$

Assuming that the serial-chain is canonical, the symmetric positive semi-definite  $Y(k)$  diagonal matrices can be computed via the following  $O(\mathcal{N})$  base-to-tip scatter recursion:

$$\left\{ \begin{array}{l} Y(n+1)=\mathbf{0} \\ \text{for } k=1 \cdots n \\ Y(k)=\mathbb{A}^*(k+1, k)Y(k+1)\mathbb{B}(k+1, k)+X(k) \\ \text{end loop} \end{array} \right. \quad (29)$$

While  $Y$  defines the block-diagonal elements of  $Z$ , the following recursive expressions describe all the terms, including the off-diagonal ones:

$$Z(i, j)=\left\{ \begin{array}{ll} Y(i) & \text{for } i=j \\ \mathbb{A}^*(i+1, i)Z(i+1, j) & \text{for } i < j \\ Z(i, j+1)\mathbb{B}(j+1, j) & \text{for } i > j \end{array} \right. \quad (30)$$

**Proof:** Unlike trees, all nodes in a serial-chain systems are related, i.e., every pair of nodes is connected by a directed path. Hence,  $R = \mathbf{0}$  in Eq. 21 for these systems. Substituting this in Eq. 21, and remembering that  $\mathcal{E}_A^* Y \mathcal{E}_B$  in Eq. 22 is block-diagonal for serial-chains leads to this result.

Observe that Step (4) in Algorithm 4.3 is not needed for serial-chain systems.

## 5 Riccati equations for SKO models

This section studies another type of SPO quadratic equation, known as a Riccati equation, for SKO models.

**Lemma 7 The Riccati equation for SKO models**—Let  $(H, \mathbb{A}, \mathbf{M})$  denote spatial operators for an SKO model. The following Riccati equation has a block-diagonal, symmetric and positive-definite operator solution,  $\mathcal{P}$ :

$$M = \mathcal{P} - \mathcal{E}_A \left[ \mathcal{P} - \mathcal{P} H^* (H \mathcal{P} H^*)^{-1} H \mathcal{P} \right] \mathcal{E}_A^* \quad (31)$$

The expression in Eq. 31 can be broken down into simpler sub-expressions as follows:

$$M = \mathcal{P} - \mathcal{E}_A \left[ \underbrace{\mathcal{P} - \mathcal{P} H^* (H \mathcal{P} H^*)^{-1} H \mathcal{P}}_{\mathcal{D}} \right] \mathcal{E}_A^*$$

$$\underbrace{\hspace{10em}}_{\mathcal{G}}$$

$$\underbrace{\hspace{10em}}_{\tau}$$

$$\underbrace{\hspace{10em}}_{\mathcal{P}^+}$$

The above sub-expressions define the block-diagonal spatial operators:

$$\begin{aligned} \mathcal{D} &\triangleq H \mathcal{P} H^*, & \mathcal{G} &\triangleq \mathcal{P} H^* \mathcal{D}^{-1}, & \tau &\triangleq \mathcal{G} H \\ \bar{\tau} &\triangleq \mathbf{I} - \tau, & \mathcal{P}^+ &\triangleq \bar{\tau} \mathcal{P} = \bar{\tau} \mathcal{P} \bar{\tau}^* \end{aligned} \quad (32)$$

The  $\mathcal{P}(k)$  and other diagonal elements can be computed by the following  $O(N)$  tips-to-base gather recursion:

$$\left\{ \begin{array}{l} \text{for all nodes } k \text{ (tips - to - base gather)} \\ \mathcal{P}(k) = \sum_{j \in \mathcal{C}(k)} \mathbb{A}(k, j) \mathcal{P}^+(j) \mathbb{A}^*(k, j) + \mathbf{M}(k) \\ \mathcal{D}(k) = H(k) \mathcal{P}(k) H^*(k) \\ \mathcal{G}(k) = \mathcal{P}(k) H^*(k) \mathcal{D}^{-1}(k) \\ \tau(k) = \mathcal{G}(k) H(k) \\ \bar{\tau}(k) = \mathbf{I} - \tau(k) \\ \mathcal{P}^+(k) = \bar{\tau}(k) \mathcal{P}(k) \\ \text{end loop} \end{array} \right. \quad (33)$$

**Proof:** Eq. 33 is essentially a component level restatement of Eq. 31. The positive definiteness of  $\mathbf{M}$  and full row-rankness of  $H$  ensures that  $\mathcal{D}^{-1}(k)$  exists and is well defined.

### 5.1 The $\mathcal{E}_\psi$ and $\psi$ SKO and SPO operators

**Lemma 8 The  $\mathcal{E}_\psi$  SKO operator**—Let  $\mathcal{E}_\mathbb{A}$  be an SKO operator, and  $H, \mathbf{M}$ , be compatible operators satisfying the conditions in Lemma 7. Let  $\mathcal{P}$  denote the block-diagonal solution to the Eq. 31 Riccati equation. Define

$$\mathcal{E}_\psi \triangleq \mathcal{E}_\mathbb{A} \bar{\tau} \quad (34)$$

Then  $\mathcal{E}_\psi$  is an SKO operator, with weights and SPO operator  $\psi$ , defined by

$$\psi(\varphi(k), k) \triangleq \mathbb{A}(\varphi(k), k) \bar{\tau}(k), \quad \text{and} \quad \psi \triangleq (\mathbf{I} - \mathcal{E}_\psi)^{-1} \quad (35)$$

Moreover, the Eq. 31 Riccati equation can be re-expressed as:

$$\mathbf{M} = \mathcal{P} - \mathcal{E}_\psi \mathcal{P} \mathcal{E}_\psi^* = \mathcal{P} - \mathcal{E}_\mathbb{A} \mathcal{P} \mathcal{E}_\mathbb{A}^* \quad (36)$$

**Proof:** Since  $\mathcal{E}_\mathbb{A}$  is an SKO operator, we have

$$\begin{aligned} \mathcal{E}_\psi &\stackrel{34}{=} \left( \sum_{k=1}^n e_{\varphi(k)} \mathbb{A}(\varphi(k), k) e_k^* \right) \bar{\tau} = \sum_{k=1}^n e_{\varphi(k)} \mathbb{A}(\varphi(k), k) \bar{\tau}(k) e_k^* \\ &\stackrel{35}{=} \sum_{k=1}^n e_{\varphi(k)} \psi(\varphi(k), k) e_k^* \end{aligned}$$

This implies that  $\mathcal{E}_\psi$  is indeed an SKO operator, with weights defined by  $\psi(\mathcal{P}(k), k)$ . Thus, its 1-resolvent,  $\psi$ , defined by Eq. 35 exists.

The first part of the Riccati equation expression in Eq. 36. This is equivalent to Eq. 31 follows from the direct use of the expressions in Eq. 32 to observe that:

$$\mathcal{P} - \mathcal{P} H^* (H \mathcal{P} H^*)^{-1} H \mathcal{P} = \bar{\tau} \mathcal{P} = \mathcal{P} \bar{\tau}^* = \bar{\tau} \mathcal{P} \bar{\tau}^* = \mathcal{P}^+$$

Unlike the  $\mathcal{E}_\mathbb{A}$  SKO operators, which are defined during the SKO model formulation process, the  $\mathcal{E}_\psi$  SKO operators are derived operators obtained from the solution to the Riccati equation. Likewise, the  $\psi$  SPO operator is also a derived by-product of the solution of the

Riccati equation. Observe that the  $\psi(\mathcal{P}(k), k)$  matrices are singular even when  $\mathbb{A}(\mathcal{P}(k), k)$  is non-singular.

## 5.2 Operator identities

Define the  $\mathcal{G}$  spatial operator as follows:

$$\mathcal{K} = \mathcal{E}_\psi \mathcal{G} \quad (37)$$

$\mathcal{G}$  is *not* block-diagonal, but has structure similar to that of  $\mathcal{E}_\mathbb{A}$ . The following lemma establishes operator identities that will be needed later.

### Lemma 9 Useful spatial operator identities

1.

$$\begin{aligned} H\mathcal{G} &= I, & H\tau &= H, & H\bar{\tau} &= \mathbf{0} \\ I + H\mathbb{A}\mathcal{K} &= H\mathbb{A}\mathcal{G} \end{aligned} \quad (38)$$

2.

$$H\mathbb{A}\bar{\tau} = H\mathbb{A}\mathcal{E}_\psi \quad (39)$$

3.

$$H\mathbb{A}\tau = H\mathbb{A}(I - \mathcal{E}_\psi) = H\mathbb{A}\psi^{-1} \quad (40)$$

4.  $\mathbb{A}M\psi^*$  has the following disjoint decomposition:

$$\mathbb{A}M\psi^* = \mathcal{P} + \tilde{\mathbb{A}}\mathcal{P} + \mathcal{P}\tilde{\psi}^* \quad (41)$$

### Proof

1. The expressions in Eq. 38 follow from Eq. 32 and Eq. 37.

2. We have

$$H\mathbb{A}\bar{\tau} = H(\tilde{\mathbb{A}} + I)\bar{\tau} = H\mathbb{A}\mathcal{E}_\psi \bar{\tau} + H\bar{\tau} \stackrel{34,32}{=} H\mathbb{A}\mathcal{E}_\psi$$

3. We have

$$H\mathbb{A}\tau \stackrel{32}{=} H\mathbb{A}(I - \bar{\tau}) \stackrel{39}{=} H\mathbb{A}(I - \mathcal{E}_\psi) \stackrel{35}{=} H\mathbb{A}\psi^{-1}$$

4. Pre and post multiplying Eq. 36 by  $\mathbb{A}$  and  $\psi^*$ , then simplifying, leads to Eq. 41.

The following lemma derives additional operator identities.

### Lemma 10 Additional spatial operator identities

1.

$$\psi M\psi^* = \mathcal{P} + \tilde{\psi}\mathcal{P} + \mathcal{P}\tilde{\psi}^* \quad (42)$$

2.

$$\psi^{-1} - \mathbb{A}^{-1} = \mathcal{K}H \quad (43)$$



$$\begin{aligned}
 3. \quad & \psi^{-1}\mathbb{A} = \mathbf{I} + \mathcal{K}H\mathbb{A} \\
 & \mathbb{A}\psi^{-1} = \mathbf{I} + \mathbb{A}\mathcal{K}H \\
 & \mathbb{A}^{-1}\psi = \mathbf{I} - \mathcal{K}H\psi \\
 & \psi\mathbb{A}^{-1} = \mathbf{I} - \psi\mathcal{K}H
 \end{aligned} \tag{44}$$

$$\begin{aligned}
 4. \quad & [\mathbf{I} - H\psi\mathcal{K}]H\mathbb{A} = H\psi \\
 & \mathbb{A}\mathcal{K}[\mathbf{I} - H\psi\mathcal{K}] = \psi\mathcal{K} \\
 & [\mathbf{I} + H\mathbb{A}\mathcal{K}]H\psi = H\mathbb{A} \\
 & \psi\mathcal{K}[\mathbf{I} + H\mathbb{A}\mathcal{K}] = \mathbb{A}\mathcal{K}
 \end{aligned} \tag{45}$$

$$5. \quad H\psi M\psi^* H^* = \mathcal{D} \tag{46}$$

$$6. \quad \mathbb{A}M\mathbb{A}^* = \mathcal{P} + \tilde{\mathbb{A}}\mathcal{P} + \mathcal{P}\tilde{\mathbb{A}}^* + \mathbb{A}\mathcal{K}\mathcal{D}\mathcal{K}^*\mathbb{A}^* \tag{47}$$

$$7. \quad \mathbb{A}M\mathbb{A}^* H^* = [\mathbf{I} + \mathbb{A}\mathcal{K}H]\mathcal{P}\mathbb{A}^* H^* \tag{48}$$

**Proof**

1. Pre- and post-multiplying Eq. 36  $\psi$  by and  $\psi^*$ , we obtain

$$\begin{aligned}
 \psi M\psi^* &= \psi\mathcal{P}\psi^* - \psi\mathcal{E}_\psi\mathcal{P}\mathcal{E}_\psi^*\psi^* = (\tilde{\psi} + \mathbf{I})\mathcal{P}(\tilde{\psi} + \mathbf{I})^* - \tilde{\psi}\mathcal{P}\tilde{\psi}^* \\
 &= \mathcal{P} + \tilde{\psi}\mathcal{P} + \mathcal{P}\tilde{\psi}^*
 \end{aligned}$$

2. We have

$$\begin{aligned}
 \psi^{-1} &= \mathbf{I} - \mathcal{E}_\psi \stackrel{32}{=} \mathbf{I} - \mathcal{E}_\mathbb{A}\bar{\tau} \stackrel{32}{=} (\mathbf{I} - \mathcal{E}_\mathbb{A}) + \mathcal{E}_\mathbb{A}\tau \\
 &\stackrel{32}{=} \mathbb{A}^{-1} + \mathcal{E}_\mathbb{A}\mathcal{G}H \stackrel{32}{=} \mathbb{A}^{-1} + \mathcal{K}H
 \end{aligned}$$

3. Pre- and post-multiplying Eq. 43 by  $\mathbb{A}$  leads to the first pair of identities in Eq. 44. Repeating the process using  $\psi$  leads to the latter pair.

4. We have

$$[\mathbf{I} - H\psi\mathcal{K}]H\mathbb{A} = H[\mathbf{I} - \psi\mathcal{K}H]\mathbb{A} \stackrel{44}{=} H(\psi\mathbb{A}^{-1})\mathbb{A} = H\psi$$

Similar use of the other identities in Eq. 44 leads to the the remaining identities in Eq. 45.

5. We have

$$\begin{aligned}
 H\psi M\psi^* H^* &\stackrel{36}{=} H(\mathcal{P} + \tilde{\psi}\mathcal{P} + \mathcal{P}\tilde{\psi}^*)H^* \\
 &\stackrel{32}{=} \mathcal{D} + H\tilde{\psi}\mathcal{P}H^* + H\mathcal{P}\tilde{\psi}^*H^* \\
 &= \mathcal{D} + H\psi\mathcal{E}_\psi\mathcal{P}H^* + H\mathcal{P}\mathcal{E}_\psi^*\psi^*H^* \\
 &\stackrel{32}{=} \mathcal{D} + H\psi\mathcal{E}_\psi\tilde{\mathcal{P}}H^* + H\mathcal{P}\tilde{\mathcal{P}}^*\mathcal{E}_\psi^*\psi^*H^* \\
 &\stackrel{32}{=} \mathcal{D} + H\psi\mathcal{E}_\psi\mathcal{P}^+H^* + H\mathcal{P}^+\mathcal{E}_\psi^*\psi^*H^* \\
 &\stackrel{38,32}{=} \mathcal{D}
 \end{aligned}$$

6. Now

$$\begin{aligned}
 \mathbb{A}M\mathbb{A}^* &= (\mathbb{A}\psi^{-1})\psi M\mathbb{A}^* \stackrel{41}{=} (\mathbb{A}\psi^{-1})(\psi\mathcal{P} + \mathcal{P}\tilde{\mathbb{A}}^*) \\
 &\stackrel{43}{=} \mathbb{A}\mathcal{P} + (\mathbb{I} + \mathbb{A}\mathcal{K}H)\mathcal{P}\tilde{\mathbb{A}}^* = \mathbb{A}\mathcal{P} + \mathcal{P}\tilde{\mathbb{A}}^* + \tilde{\mathbb{A}}\mathcal{K}H\mathcal{P}\tilde{\mathbb{A}}^* \\
 &\stackrel{32,37}{=} \mathcal{P} + \tilde{\mathbb{A}}\mathcal{P} + \mathcal{P}\mathbb{A}^* + \mathbb{A}\mathcal{K}\mathcal{D}\mathcal{K}^*\mathbb{A}^*
 \end{aligned}$$

7. We have

$$\begin{aligned}
 \mathbb{A}M\mathbb{A}^*H^* &= (\mathbb{A}\psi^{-1})(\psi M\mathbb{A}^*)H^* \stackrel{41}{=} (\mathbb{A}\psi^{-1})(\tilde{\psi}\mathcal{P} + \mathcal{P}\mathbb{A}^*)H^* \\
 &= \mathbb{A}\mathcal{E}_\psi\mathcal{P}H^* + (\mathbb{A}\psi^{-1})\mathcal{P}\mathbb{A}^*H^* \\
 &\stackrel{32}{=} \mathbb{A}\mathcal{E}_\psi\mathcal{P}^+H^* + (\mathbb{I} + \mathbb{A}\mathcal{K}H)\mathcal{P}\mathbb{A}^*H^* \\
 &\stackrel{38}{=} (\mathbb{I} + \mathbb{A}\mathcal{K}H)\mathcal{P}\mathbb{A}^*H^*
 \end{aligned}$$

It is noteworthy that the  $H\psi M\psi^*H^*$  expression in Eq. 46 is block-diagonal, in contrast to the similar expression,  $\mathcal{M} = H \mathbb{A}M \mathbb{A}^*H^*$ , for the non-diagonal mass matrix.

## 6 SKO model mass matrix factorization and inversion

The following lemma derives the mass matrix factorization and inversion properties for SKO models.

**Lemma 11 SKO mass matrix factorization and inversion**—Let  $(H, \mathbb{A}, M)$  denote spatial operators for an SKO model. Recall that its mass-matrix is defined as  $\mathcal{M} = H \mathbb{A}M \mathbb{A}^* H^*$ .

1. The  $\mathcal{M}$  mass matrix has an alternative Innovations Operator factorization defined by

$$\mathcal{M} = [\mathbb{I} + H\mathbb{A}\mathcal{K}]\mathcal{D}[\mathbb{I} + H\mathbb{A}\mathcal{K}]^* \tag{49}$$

In this equation,  $\mathcal{D}$  and  $\mathcal{K}$  are operators obtained from the solution of the discrete Riccati equation from Lemma 8. for the SKO model.

2.  $[\mathbb{I} + H \mathbb{A} \mathcal{K}]$  is invertible, with inverse given by:

$$[\mathbb{I} + H\mathbb{A}\mathcal{K}]^{-1} = [\mathbb{I} - H\psi\mathcal{K}] \tag{50}$$

3. The mass matrix  $\mathcal{M}$  is invertible, and the expression for its inverse is

$$\mathcal{M}^{-1} = [\mathbb{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [\mathbb{I} - H\psi\mathcal{K}] \tag{51}$$

**Proof**

1. We have

$$\begin{aligned} \mathcal{M} &= H \mathbb{A} \mathcal{M} \mathbb{A}^* H^* = H(\mathbb{A} \psi^{-1}) \psi \mathcal{M} \psi^* (\mathbb{A} \psi^{-1})^* H^* \\ &\stackrel{44}{=} H[\mathbf{I} + \mathbb{A} \mathcal{K} H] \psi \mathcal{M} \psi^* [\mathbf{I} + \mathbb{A} \mathcal{K} H]^* H^* \\ &= [\mathbf{I} + H \mathbb{A} \mathcal{K}] (H \psi \mathcal{M} \psi^* H^*) [\mathbf{I} + H \mathbb{A} \mathcal{K}]^* \\ &\stackrel{46}{=} [\mathbf{I} + H \mathbb{A} \mathcal{K}] \mathcal{D} [\mathbf{I} + H \mathbb{A} \mathcal{K}]^* \end{aligned}$$

2. From the standard matrix identity  $(\mathbf{I} + AB)^{-1} = \mathbf{I} - A(\mathbf{I} + BA)^{-1}B$ , we have:

$$\begin{aligned} [\mathbf{I} + H \mathbb{A} \mathcal{K}]^{-1} &= \mathbf{I} - H[\mathbf{I} + \mathbb{A} \mathcal{K} H]^{-1} \mathbb{A} \mathcal{K} \stackrel{44}{=} \mathbf{I} - H(\mathbb{A} \psi^{-1})^{-1} \mathbb{A} \mathcal{K} \\ &= \mathbf{I} - H \psi \mathcal{K} \end{aligned}$$

3. We have

$$\begin{aligned} \mathcal{M}^{-1} &\stackrel{49}{=} \{[\mathbf{I} + H \mathbb{A} \mathcal{K}] \mathcal{D} [\mathbf{I} + H \mathbb{A} \mathcal{K}]^*\}^{-1} = [\mathbf{I} + H \mathbb{A} \mathcal{K}]^{-*} \mathcal{D}^{-1} [\mathbf{I} + H \mathbb{A} \mathcal{K}]^{-1} \\ &\stackrel{50}{=} [\mathbf{I} - H \psi \mathcal{K}]^* \mathcal{D}^{-1} [\mathbf{I} - H \psi \mathcal{K}]^{-1} \end{aligned}$$

**Corollary 6.1 Determinant of the mass matrix**

1. Show that  $[\mathbf{I} + H \mathbb{A} \mathcal{K}]$  and  $[\mathbf{I} - H \psi \mathcal{K}]$  are strictly lower triangular for canonical trees, and that they have identity block matrices along the diagonal.
2. Show that the determinant of the Newton-Euler operator factor  $[\mathbf{I} + H \mathbb{A} \mathcal{K}]$  is

$$\det\{\mathbf{I} + H \mathbb{A} \mathcal{K}\} = 1 \tag{52}$$

3. Furthermore, show that the determinant of the mass matrix is given by

$$\det\{\mathcal{M}\} = \prod_{k=1}^n \det\{\mathcal{D}(k)\} \tag{53}$$

**Proof**

1. We have

$$\mathbf{I} - H \mathbb{A} \mathcal{K} \stackrel{37}{=} \mathbf{I} - H \mathbb{A} \mathcal{E}_A \mathcal{G} = \mathbf{I} - H \tilde{\mathbb{A}} \mathcal{G}$$

For a canonical tree,  $\tilde{\mathbb{A}}$  is strictly lower triangular, while  $H$  and  $\mathcal{G}$  are block-diagonal. Hence,  $[\mathbf{I} + H \mathbb{A} \mathcal{K}]$  is lower triangular with identity blocks along the diagonal. Moreover, from Eq. 50 we know that  $[\mathbf{I} - H \psi \mathcal{K}]$  is its inverse. From matrix theory, we know that the inverse of a lower-triangular matrix is also lower-triangular, and that the diagonal elements are inverses of each other. It thus, follows that for a canonical tree,  $[\mathbf{I} - H \psi \mathcal{K}]$  is also lower-triangular with identity blocks along its diagonal.

2. For canonical trees, the above part established that  $[\mathbf{I} + H \mathbb{A} \mathcal{K}]$  is a lower-triangular matrix with identity matrices along the diagonal. Since, the determinant of a lower-triangular matrix is the product of the determinants of the block elements along its diagonal, it follows that Eq. 52 holds for canonical trees. Since

all tree can be converted into canonical trees by a simple renumbering of the bodies, there exists a permutation matrix which transforms the Newton-Euler factors for a tree into the corresponding factors for a canonical version of the tree. Since permutation matrices are orthogonal, their determinants are 1, and hence, the determinant of the canonical and non-canonical versions of the Newton-Euler factors are equal to each other and are both 1. This establishes Eq. 52.

3. For Eq. 53, we have

$$\begin{aligned} \det\{\mathcal{M}\} &\stackrel{49}{=} \det\{[\mathbf{I}+H\mathbb{A}\mathcal{K}]\mathcal{D}[\mathbf{I}+H\mathbb{A}\mathcal{K}]^*\} \\ &= \det\{[\mathbf{I}+H\mathbb{A}\mathcal{K}]\}\det\{\mathcal{D}\}\det\{[\mathbf{I}+H\mathbb{A}\mathcal{K}]^*\} \\ &\stackrel{52}{=} \det\{\mathcal{D}\} = \prod_{k=1,n} \det\{\mathcal{D}(k)\} \end{aligned}$$

### 6.1 $\mathcal{O}(\mathcal{N})$ AB forward dynamics

The forward dynamics problem consists of computing the  $\ddot{\theta}$  generalized accelerations, given the  $\mathcal{T}$  generalized forces for the system. The following lemma derives an explicit operator expression for  $\ddot{\theta}$ .

**Lemma 12 Expression for  $\ddot{\theta} = \mathbf{m}^{-1}(\mathcal{T} - c)$** —The explicit expression for  $\ddot{\theta}$  is

$$\ddot{\theta} = \mathcal{M}^{-1}(\mathcal{T} - c) = [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}[\mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathbf{a} + \mathbf{b})] - \mathcal{K}^* \psi^* \mathbf{a} \tag{54}$$

**Proof:** We have

$$\begin{aligned} \ddot{\theta} &\stackrel{51}{=} [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}[\mathbf{I} - H\psi\mathcal{K}](\mathcal{T} - c) \\ &\stackrel{2}{=} [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}[\mathbf{I} - H\psi\mathcal{K}](\mathcal{T} - H\mathbb{A}(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b})) \\ &= [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}[\mathbf{I} - H\psi\mathcal{K}]\mathcal{T} \\ &\quad - [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1}[\mathbf{I} - H\psi\mathcal{K}]H\mathbb{A}(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b}) \end{aligned} \tag{55}$$

Now,

$$\begin{aligned} [\mathbf{I} - H\psi\mathcal{K}]H\mathbb{A}(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b}) &= H[\mathbf{I} - \psi\mathcal{K}H]\mathbb{A}(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b}) \\ &\stackrel{44}{=} H(\psi\mathbb{A}^{-1})\mathbb{A}(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b}) \\ &= H\psi(\mathbf{M}\mathbb{A}^* \mathbf{a} + \mathbf{b}) \\ &\stackrel{41}{=} H((\psi\mathcal{P} + \mathcal{P}\tilde{\mathbb{A}}^*)\mathbf{a} + \psi\mathbf{b}) \\ &\stackrel{32}{=} (H\psi\mathcal{P} + \mathcal{D}\mathcal{K}^*\mathbb{A}^*)\mathbf{a} + H\psi\mathbf{b} \end{aligned} \tag{56}$$

Substituting this expression into the second half of Eq. 55, it follows that

$$\begin{aligned}
& [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [\mathbf{I} - H\psi\mathcal{K}] H \mathbb{A} (M \mathbb{A}^* \mathbf{a} + \mathbf{b}) \\
& = [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [(H\psi\mathcal{P} + \mathcal{D}\mathcal{K}^* \mathbb{A}^*) \mathbf{a} + H\psi\mathbf{b}] \\
& \stackrel{44}{=} [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [H\psi\mathcal{P}\mathbf{a} + H\psi\mathbf{b}] + \mathcal{K}^* (\mathbb{A}^{-1} \psi)^* \mathbb{A}^* \mathbf{a} \\
& \stackrel{44}{=} [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} H\psi(\mathcal{P}\mathbf{a} + \mathbf{b}) + \mathcal{K}^* \psi^* \mathbf{a}
\end{aligned} \tag{57}$$

Substituting this expression in Eq. 55 leads to Eq. 54.

We group together sub-expressions in Eq. 54 to define intermediate quantities as shown below:

$$\ddot{\boldsymbol{\theta}} = [\mathbf{I} - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [\underbrace{\mathcal{T} - H\psi(\underbrace{\mathcal{K}\mathcal{T} + \mathcal{P}\mathbf{a} + \mathbf{b}}_{\mathfrak{z}})}_{\underbrace{\epsilon}_{\nu}}] - \mathcal{K}^* \psi^* \mathbf{a}$$

Using these intermediate quantities, and simplifications Eq. 54 can be re-expressed as

$$\mathfrak{z} = \mathcal{E}_\varphi \mathfrak{z}^+ + \mathcal{P}\mathbf{a} + \mathbf{b} \tag{58a}$$

$$\mathfrak{z}^+ = \mathfrak{z} + \mathcal{G}\epsilon \tag{58b}$$

$$\epsilon = \mathcal{T} - H\mathfrak{z} \tag{58c}$$

$$\nu = \mathcal{D}^{-1} \epsilon \tag{58d}$$

$$\alpha^+ = \mathcal{E}_\varphi^* \alpha \tag{58e}$$

$$\ddot{\boldsymbol{\theta}} = \nu - \mathcal{G}^* \alpha^+ \tag{58f}$$

$$\alpha = \alpha^+ + H^* \ddot{\boldsymbol{\theta}} + \mathbf{a} \tag{58g}$$

Recall that  $\psi$  is an SPO operator. We can thus use Lemmas 1 and 2 to develop a recursive algorithm for evaluating  $\ddot{\boldsymbol{\theta}}$  from the expressions in Eq. 58. The resulting  $O(\mathcal{N})$  AB forward dynamics algorithm for SKO models is described in Algorithm 6.1. The algorithm consists

of a tips-to-base gather sweep to compute the residual terms, followed by a base-to-tips scatter sweep for the generalized accelerations.

### Algorithm 6.1 $\mathcal{O}(N)$ AB forward dynamics for SKO models

1. Compute the articulated body inertia  $\mathcal{P}$ ,  $\mathcal{D}$ , etc., quantities using the gather algorithm in Eq. 33.
2. Use the following gather and scatter recursions to compute  $\bar{\theta}$ :

$$\left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (tips-to-base gather)} \\ \quad \ddot{\mathbf{z}}(\mathbf{k}) = \sum_{\forall j \in \mathcal{B}(\mathbf{k})} \mathbb{A}(\mathbf{k}, j) \ddot{\mathbf{z}}^+(j) + \mathcal{P}(\mathbf{k}) \mathbf{a}(\mathbf{k}) + \mathbf{b}(\mathbf{k}) \\ \quad \boldsymbol{\epsilon}(\mathbf{k}) = \mathcal{T}(\mathbf{k}) - H(\mathbf{k}) \ddot{\mathbf{z}}(\mathbf{k}) \\ \quad \boldsymbol{\nu}(\mathbf{k}) = \mathcal{D}^{-1}(\mathbf{k}) \boldsymbol{\epsilon}(\mathbf{k}) \\ \quad \ddot{\mathbf{z}}^+(\mathbf{k}) = \ddot{\mathbf{z}}(\mathbf{k}) + \mathcal{G}(\mathbf{k}) \boldsymbol{\epsilon}(\mathbf{k}) \\ \text{end loop} \\ \\ \left\{ \begin{array}{l} \text{for all nodes } \mathbf{k} \text{ (base-to-tips scatter)} \\ \quad \alpha^+(\mathbf{k}) = \mathbb{A}^*(\varphi(\mathbf{k}), \mathbf{k}) \alpha(\varphi(\mathbf{k})) \\ \quad \ddot{\boldsymbol{\theta}}(\mathbf{k}) = \boldsymbol{\nu}(\mathbf{k}) - \mathcal{G}^*(\mathbf{k}) \alpha^+(\mathbf{k}) \\ \quad \alpha(\mathbf{k}) = \alpha^+(\mathbf{k}) + H^*(\mathbf{k}) \ddot{\boldsymbol{\theta}}(\mathbf{k}) + \mathbf{a}(\mathbf{k}) \\ \text{end loop} \end{array} \right. \end{array} \right.$$

## 6.2 Application: Tree Operational Space Inertia

Another application of the SPO techniques is for operational space inertias [12, 13] which is essentially the mass matrix of the system reflected to a few “task space” nodes on the system. These are useful for manipulation tasks involving force control. With  $\mathcal{B}$  denoting the pick-off operator for these nodes [17], and with the Jacobian matrix for these nodes  $\mathcal{J} = \mathcal{B}^* \varphi^* H^*$  [7], the computation of operational space inertia is the inverse of the  $\mathcal{J} \mathcal{M}^{-1} \mathcal{J}^*$  matrix. This can be an expensive computation to carry out. However, note that

$$\mathcal{J} \mathcal{M}^{-1} \mathcal{J}^* = \mathcal{B}^* \varphi^* H^* (\mathbf{I} - H \psi \mathcal{K})^* \mathcal{D}^{-1} (\mathbf{I} - H \psi \mathcal{K}) H \varphi \mathcal{B} \quad (59)$$

Application of Eq. 45 on page 18 then results in the simpler form:

$$\mathcal{J} \mathcal{M}^{-1} \mathcal{J}^* = \mathcal{B}^* \Omega \mathcal{B}, \quad \text{where } \Omega \triangleq \psi^* H^* \mathcal{D}^{-1} H \psi \in \mathcal{R}^{6n \times 6n} \quad (60)$$

We refer to  $\Omega$  as the **extended operational space compliance matrix**. From its definition, it is clear that  $\Omega$  is a symmetric, positive semi-definite matrix since  $\mathcal{D}^{-1}$  is a symmetric positive-definite matrix. The structure and computation of the operational space inertia thus depends on the structure and computation of  $\Omega$ .

Recalling that  $\varphi$  is an SPO operator, and that  $H^* \mathcal{D}^{-1} H$  is block-diagonal, we can apply the backwards Lyapunov equation Lemma 5 to decompose and compute elements of  $\Omega$  efficiently. We skip the details here and refer the reader to [4, 13, 18] for details.

## 7 SPO operator sparsity structure

Researchers have used system-level matrices and operators to analyze and exploit the sparsity structure of the mass matrix to develop efficient computational algorithms for the inverse and forward dynamics problems [1, 3, 14, 19]. In this section, study the sparsity structure of the SKO and SPO operators and the mass matrix for tree-topology systems.

### 7.1 Decomposition into serial-chain segments

Topologically, serial-chain systems represent the simplest examples of tree systems, and serve as useful elemental units for studying the sparsity structure of matrices for tree-topology systems. Unlike trees, all bodies in a serial-chain are *related*, so that, for any pair of bodies, one is necessarily an ancestor of the other.

Figure 3 illustrates the decomposition of a tree into branches 1 through 5, where each of the branches is a serial-chain segment. With such a decomposition, the tree system can be regarded as a new tree, with one node for each of the serial-chain branch segments. Thus, the new tree of branches is homeomorphic to the original tree<sup>3</sup>. The sparsity structure of the spatial operators (e.g., SPO, mass matrix) can be inferred directly from the structure of the tree of branches. Due to their serial-chain structure, the blocks associated with individual serial-chain branches are dense, with sparsity arising for branch pairs that are *unrelated*, i.e., are not connected by a directed path in the tree of branches.

### 7.2 Sparsity structure of the $\mathcal{E}_A$ SKO matrix

Assuming that the original tree is a canonical tree, Figure 4 and Eq. 61 illustrate the decomposition of the system-level  $\mathcal{E}_A$  SKO operator in terms of the SKO operators for each of the serial-chain branch segments.

$$\mathcal{E}_A = \begin{pmatrix} \mathcal{E}_{A_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{E}_{A_2} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{E}_{A_3} & \mathbf{0} & \mathbf{0} \\ \mathcal{E}_{A_{4,1}} & \mathcal{E}_{A_{4,2}} & \mathbf{0} & \mathcal{E}_{A_4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathcal{E}_{A_{5,3}} & \mathcal{E}_{A_{5,4}} & \mathcal{E}_{A_5} \end{pmatrix} \quad (61)$$

$\mathcal{E}_{A_j}$  denotes the SKO operator for the  $j^{\text{th}}$  branch segment. Also, when the  $k^{\text{th}}$  branch is a child of the  $j^{\text{th}}$  branch, the  $\mathcal{E}_{A_{j,k}}$  block denotes the non-zero connector block between them. All other blocks are zero. The structure of  $\mathcal{E}_{A_{j,k}}$  is as follows:

$$\mathcal{E}_{A_{j,k}} \triangleq \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbb{A}(1_j, n_k) \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (62)$$

where  $1_j$  denotes the tip body on the  $j^{\text{th}}$  branch that is the parent of the  $n_k$  base-body of the  $k^{\text{th}}$  branch.

<sup>3</sup>A pair of digraphs are said to be **homeomorphic** if they can both be obtained from a common digraph by a sequence of adding and removing nodes (also known as subdivisions) along serial segments [2, 20].

### 7.3 Sparsity structure of the $\mathbb{A}$ matrix

With  $\varphi_j \triangleq (\mathbf{I} - \epsilon_j)^{-1}$  denoting the SPO operator for the  $j^{\text{th}}$  branch, the overall structure of the system level  $\mathbb{A}$  SPO operator for the Figure 3 tree is illustrated in Figure 5 and Eq. 63:

$$\mathbb{A} = \begin{pmatrix} \mathbb{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbb{A}_2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbb{A}_3 & \mathbf{0} & \mathbf{0} \\ \mathbb{A}_{4,1} & \mathbb{A}_{4,2} & \mathbf{0} & \mathbb{A}_4 & \mathbf{0} \\ \mathbb{A}_{5,1} & \mathbb{A}_{5,2} & \mathbb{A}_{5,3} & \mathbb{A}_{5,4} & \mathbb{A}_5 \end{pmatrix} \quad (63)$$

The  $\mathbb{A}_{j,k}$  blocks denote the non-zero connector block between *related* serial-chain segments. All other blocks are zero. The general expression for the  $\mathbb{A}_{j,k}$  elements is:

$$\mathbb{A}_{j,k} = \mathbb{A}_j \mathcal{E}_{\mathbb{A}_{j,i}} \mathbb{A}_{i,k} \text{ with } i \in C(j), i \geq k$$

In the above,  $\mathbb{A}_{k,k} \equiv \mathbb{A}_k$ .

### 7.4 Sparsity structure of the $\mathbb{M}$ mass matrix

Based on the mass matrix decomposition discussed in Section 4.2, Figure 6 illustrates the sparsity structure of the mass matrix for the tree-topology system in Figure 3, partitioned according to its branch segments. As expected, the sparsity structure of the mass matrix mirrors the sparsity of the  $\mathbb{A}$  SPO operator in Figure 5 and that of its transpose. The block-diagonal of the mass matrix contains dense blocks, one for each of the branch segments in the tree. The off-diagonal blocks are zero for *unrelated* segment pairs, i.e., ones where neither is the ancestor of the other. The remaining nonzero off-diagonal elements are for segment pairs that are related. This structure of the mass matrix is a generalization of the discussion on this subject in Section 4.2. The topology dependency of the mass matrix's sparsity structure was initially described in Rodriguez et al [18]. Additional discussion on this topic can be found in Featherstone [3].

## 8 Generalized SKO formulation process

In this paper, we have derived analytical techniques and efficient algorithms for SKO models of multibody systems. These have included

- Recursive  $O(\mathcal{N})$  procedures for computing SPO operator and stacked vector products.
- General  $O(\mathcal{N})$  Newton-Euler inverse dynamics algorithms.
- Solutions for the forward Lyapunov equations and decomposition of  $\mathbb{A}^* \mathbb{X} \mathbb{B}^*$  operator product.
- General  $O(\mathcal{N}^2)$  algorithm for computing the mass matrix.
- Solution for the backward Lyapunov equations and decomposition of  $\mathbb{A}^* \mathbb{X} \mathbb{B}$  operator product.
- Recursive algorithms for computing the  $\mathbb{A}^* \mathbb{X} \mathbb{B}$  operator product.
- The general  $O(\mathcal{N})$  articulated body inertia solution of the Riccati equation.
- Several operator identities.
- The alternative Innovations Operator Factorization of the mass matrix.



- An analytical expression for the inverse of the mass matrix.
- The analytical expression for the determinant of the mass matrix.
- The general  $O(N)$  AB forward dynamics algorithm.

The above results required no assumptions on the SKO model regarding the SKO weight matrices, the components of the other spatial operators, or the structure of the tree digraph. Thus, any multibody system formulation satisfying the requirements of the SKO model has available to it the full spectrum of these techniques and efficient algorithms. With this as motivation, we outline the general steps in the development of an SKO multibody dynamics formulation:

**Develop an SKO model:** The key starting point of an SKO formulation is the development of an SKO model for the system. A systematic procedure for this is deferred till Section 8.1.

**Apply SKO techniques:** Once the SKO model has been developed, the analytical results and efficient algorithms for spatial operators described above can be applied to the system.

**Optimize algorithms:** Finally, optimizations that take advantage of the specific features of the SKO model can be further applied to the SKO algorithms. Though the SKO algorithms are already highly efficient, there are usually several opportunities for further optimizations based on the specific structure, sparsity, and redundancies of the SKO weight matrices, joint map matrices, etc. Even though such system specific optimization are applied only in this last step, they can significantly transform the structure of the eventual algorithms.

### 8.1 Procedure for developing an SKO model

Outlined below is a procedure for developing an SKO model for general multibody systems. It provides guidelines for identifying the SKO weight matrices and the components of the  $H$  and  $M$  operators. Figure 7 illustrates the key steps in the procedure.

- 1. Identify system tree digraph:** First, identify a tree digraph for the SKO model. For tree-topology multibody systems, this is usually straightforward, with the standard tree digraph for the system being a good candidate.  
  
However, there is no requirement that the nodes in the tree digraph be in one-to-one correspondence with the physical bodies in the system, as is the case for the standard tree digraph.
- 2. Node equations of motion:** Establish the equations of motion of the component nodes in the system. To accomplish this, identify the  $v(k)$  velocities for the component nodes. The size of the velocity vector for a node determines the  $m_k$  BWA weight dimension for the node. Also, identify the appropriate  $m_k \times m_k$  dimensional  $M(k)$  inertia matrix for each node. Together with the node velocity, the  $\frac{1}{2}v^*(k)M(k)v(k)$  expression should define the kinetic energy contribution of the node.
- 3. Identify inter-node velocity relationships:** Identify the recursive relationship between the  $v(k)$  velocity of a node and that of its parent node, and the  $\dot{\theta}(k)$  generalized velocities of the hinge connecting them. This will help identify the parent/child SKO weight matrix,  $\Delta(\mathcal{P}(k), k)$ , and the component  $H(k)$  joint map matrices associated with the connecting hinge. The  $\Delta^*(\mathcal{P}(k), k)$  term defines the contribution of the  $v(\mathcal{P}(k))$  parent node's velocity to  $v(k)$  while  $H^*(k)$  defines the contribution from the  $\dot{\theta}(k)$  generalized velocities.

The hinges connecting adjacent nodes are not required to be physical hinges. This is especially true in cases where multiple bodies are assigned to nodes in the digraph.

4. **Assemble Newton-Euler mass matrix expression:** Assemble the system-level stacked vectors for the hinge generalized velocities, the node velocities, etc., and the various SKO, SPO, etc., spatial operators leading to the system-level equations of motion, the Newton-Euler factorization of the  $\mathcal{M}$  mass matrix and the  $\mathcal{C}$  nonlinear Coriolis and velocity terms vector in Eq. 1 through Eq. 3.

## 8.2 Potential non-tree topology generalizations

The digraph in an SKO model is required to be a directed tree. We address here the potential extension of an SKO model to work with non-tree digraphs. First, we recall key steps in the SKO model development:

1. set up of the implicit relationship

$$\mathcal{V} = \mathcal{E}_{\mathbb{A}}^* \mathcal{V} + H^* \dot{\theta}$$

for the  $\mathcal{V}$  node velocities using independent generalized hinge velocity coordinates, and

2. use the nilpotency of the  $\mathcal{E}_{\mathbb{A}}$  BWA matrix to derive its  $\mathbb{A}$  1-resolvent to convert this implicit expression into the  $\mathcal{V} = \mathbb{A}^* H^* \dot{\theta}$  explicit form.

Non-tree digraphs can contain directed cycles as well as multiply-connected nodes. We now examine specific issues in developing SKO models for such non-tree digraph systems.

**Digraphs with directed cycles:** For digraphs with directed cycles, the BWA matrices are *not* nilpotent, and their 1-resolvents do not exist. Intuitively, the presence of cycles implies that there are paths of arbitrary lengths connecting nodes that are part of such cycles, since paths that loop around the cycle multiple times are legal. Thus, none of the powers of the BWA matrix, with directed cycles, are nonzero. Hence, their 1-resolvent SPO operators do not exist. This means that step (2) above breaks down, and the implicit relationship for  $\mathcal{V}$  in cannot be transformed into the explicit form.

**Multiply-connected DAGs:** Multiply-connected DAGs are digraphs containing a node with more than one parent. For these systems, the BWA matrix is still nilpotent and, hence, its 1-resolvent is well defined. The catch for such systems is the inability to identify *independent* generalized hinge velocity coordinates for the edges. Such relationships are unique and well-defined when bodies have single parents. However, when a body has multiple parents, the parent/child node velocity relationship must hold simultaneously for each parent. For them to hold simultaneously, the pair of generalized hinge velocity coordinates must be mutually consistent and, hence, they are not independent. Step (1) above therefore breaks down.

Thus, we see that the use of non-tree digraphs with SKO models has potential problems. This complicates the formulation of the dynamics of closed-chain systems. Jain [6] discuss avenues for overcoming these hurdles in order to develop and use SKO models for closed-chain systems.

## 9 Conclusions

We have identified key mathematical connections between digraph properties - adjacency matrices, nilpotency, 1-resolvents - and the spatial operators associated with the dynamics of

multibody systems. We have used these to define the abstract notions of SKO models for multibody systems, and derived a broad family of analytical results for such models.

Important illustrative examples of such analytical results include: the closed-form factorization and inversion of the system mass matrix, expressions for its determinant, factored expression for the system operational space inertia. While these results were derived in much narrower contexts in the past (e.g. rigid-body/serial-chain assumptions), the SKO model abstraction provides a unifying, abstract perspective that allows us to apply the results to the broad class of tree-topology systems where: the tree branching structure and body indexing can be arbitrary, and the weight matrices can be general, non-invertible, non-square and even of non-uniform size.

We have also shown that the SKO model analytical operator expressions can be transformed into efficient, low-order gather and scatter recursive computational algorithms whenever evaluation is required. This transformation process has been illustrated by deriving (in a general context) key computational algorithms used in the control and simulation of the robot multibody systems. Examples of such algorithms include ones for  $O(N)$  Newton-Euler inverse dynamics,  $O(N)$  AB forward dynamics,  $O(N)$  composite rigid body inertias,  $O(N^2)$  mass matrix computation, and  $O(N)$  operational space inertia computation. Once again, these algorithms apply to all SKO models and are independent of the specific nature of the component weight matrices. While these algorithms are already the most efficient known, there is opportunity for further optimization based on exploiting the system-specific structure of the weight matrices.

This paper has also studied the sparsity structure of the mass matrix and related quantities for SKO models. Such sparsity structure has been exploited in recent years to develop new families of efficient computational algorithms [3]. A systematic recipe for developing SKO models for general multibody systems has been described.

Since SKO models require tree-topology digraphs, they do not directly apply to systems with closed-graph topologies. In [6] we have begun exploring techniques for deriving SKO models for closed-graph systems by transforming the standard digraph using constraint embedding techniques [5]. Being independent of system topology and detailed structure, we expect to broaden and advance the application of the SKO principles to robotic kinematics and dynamics problems in future work. Indeed techniques such as sensitivity analysis, diagonalized dynamics formulations, under-actuated systems etc. that had previously been developed for rigid body serial-chain systems using spatial operators, we expect to generalize using SKO techniques.

## Acknowledgments

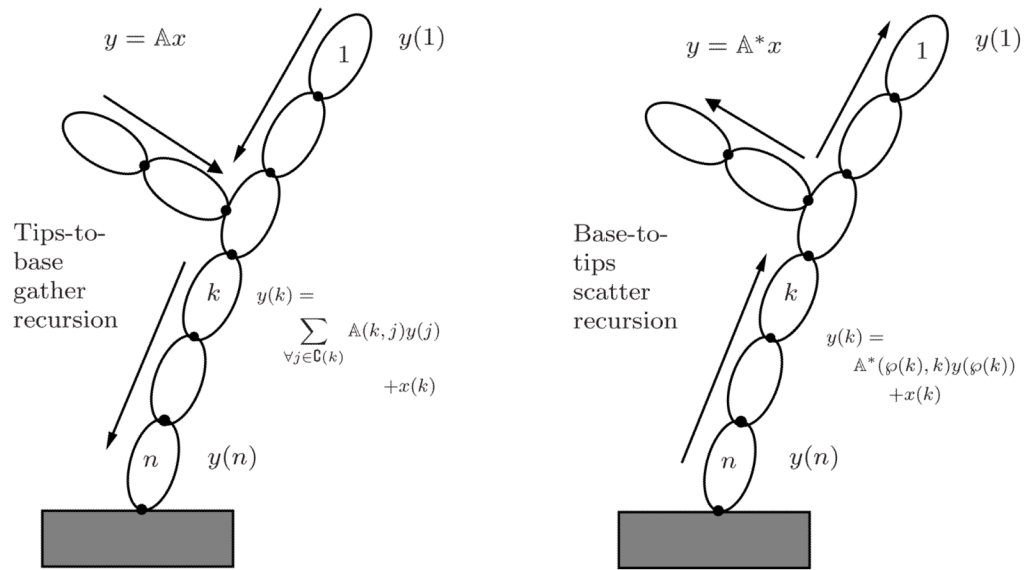
The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under contract with the National Aeronautics and Space Administration.<sup>4</sup> This project was also supported in part by Grant Number RO1GM082896-01A2 from the National Institute of Health.

## References

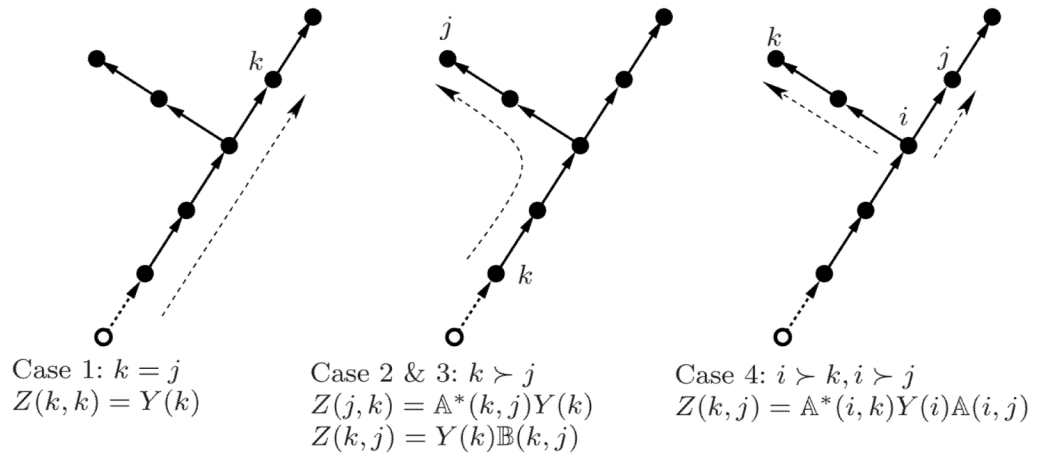
1. Baraff, D. Linear-Time Dynamics using Lagrange Multipliers. Proc. ACM SIGGRAPH '96; New Orleans. 1996. p. 137-146.
2. Cherowitzo, W. Graph and Digraph Glossary. 2009. Website, URL <http://www-math.cudenver.edu/wcherowi/courses/m4408/glossary.htm/>
3. Featherstone R. Rigid Body Dynamics Algorithms. Springer Verlag; 2008.

<sup>4</sup>© 2010 California Institute of Technology. Government sponsorship acknowledged.

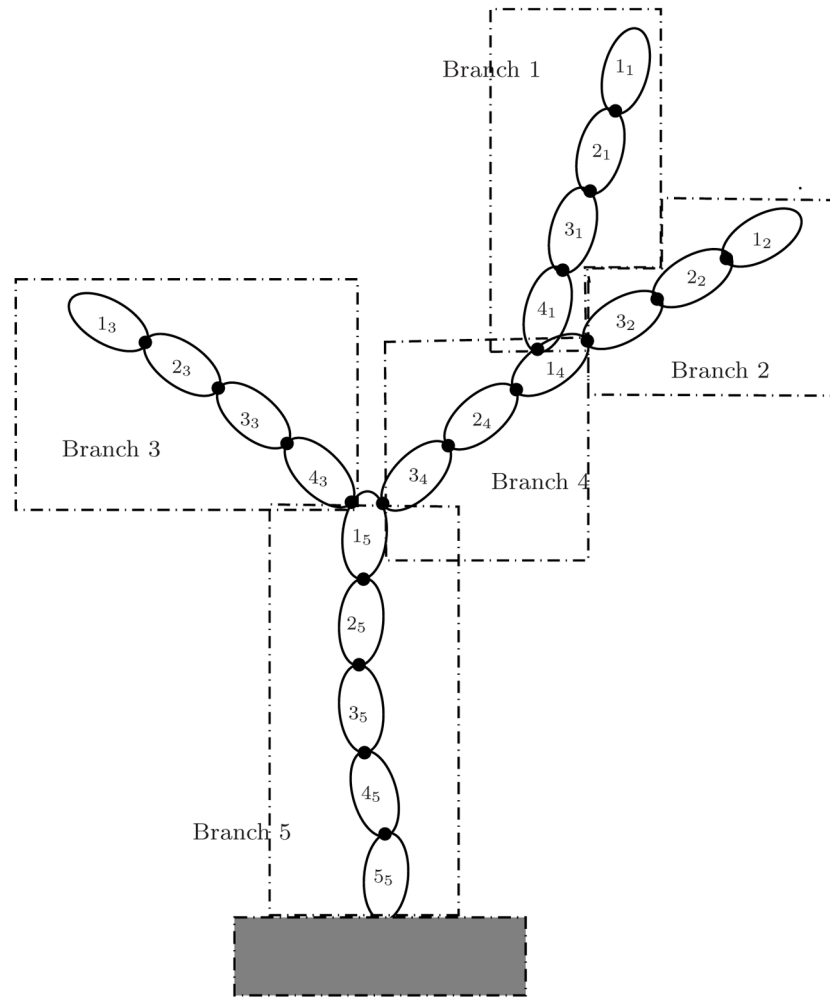
4. Jain, A. Correction to 'Spatial Operator Algebra for Multibody System Dynamics'. 2009. Website, URL <http://dartslab.jpl.nasa.gov/References/pdf/gen-mbody-correction.pdf>
5. Jain, A. Recursive Algorithms Using Local Constraint Embedding for Multi-body System Dynamics. ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference; San Diego, CA. 2009.
6. Jain A. Constraint Embedding for Robot System Dynamics. 2010 In review.
7. Jain A. Graph Theoretic Foundations of Multibody Dynamics: Part I -Structural Properties. 2010 In review.
8. Jain, A. Graph Theoretic Structure of Multibody System Spatial Operators. The 1st Joint International Conference on Multibody System Dynamics; Lappeenranta, Finland. 2010.
9. Jain, A. Robot and Multibody Dynamics: Analysis and Algorithms. Springer; New York: 2010.
10. Kane T, Levinson D. The Use of Kane's Dynamical Equations in Robotics. The International Journal of Robotics Research. 1983; 2(3):3–21.
11. Kane, T.; Levinson, D. Dynamics: Theory and Applications. McGraw-Hill; New York: 1985.
12. Khatib O. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal of Robotics and Automation. 1987; RA-3(1):43–53.
13. Kreutz-Delgado K, Jain A, Rodriguez G. Recursive Formulation of Operational Space Control. The International Journal of Robotics Research. 1992; 11(4):320–328.
14. Orlandea N, Chace M, Calahan D. A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - PART 1. ASME J Engineering for Industry. 1977; 99(2):773–779.
15. Rodriguez G. Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics. IEEE Journal of Robotics and Automation. 1987; 3(6):624–639.
16. Rodriguez G, Kreutz-Delgado K. Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix. IEEE Transactions on Robotics and Automation. 1992; 8(1):65–76.
17. Rodriguez G, Kreutz-Delgado K, Jain A. A Spatial Operator Algebra for Manipulator Modeling and Control. The International Journal of Robotics Research. 1991; 10(4):371–381.
18. Rodriguez G, Jain A, Kreutz-Delgado K. Spatial Operator Algebra for Multi-body System Dynamics. Journal of the Astronautical Sciences. 1992; 40(1):27–50.
19. Saha SK. A Decomposition of the Manipulator Inertia Matrix. IEEE Transactions on Robotics and Automation. 1997; 13(2):301–304.
20. West, DB. Introduction to Graph Theory. Prentice Hall; 2001.



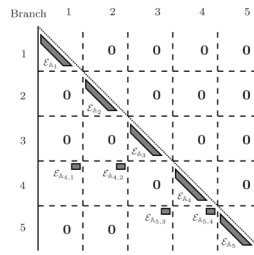
**Fig 1.** Tips-to-base gather and base-to-tips scatter recursions to evaluate  $\mathbb{A}x$  and  $\mathbb{A}^*x$ , respectively for tree-topology systems.



**Fig 2.** Illustration of the recursions for computing the  $Z(., .)$  elements for the three cases described in Algorithm 4.3 for a tree. These recursions start from the  $Y(.)$  diagonal terms.

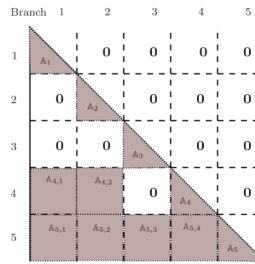


**Fig 3.** Illustration of the decomposition of a tree-topology system into a tree of serial-chain branch segments.

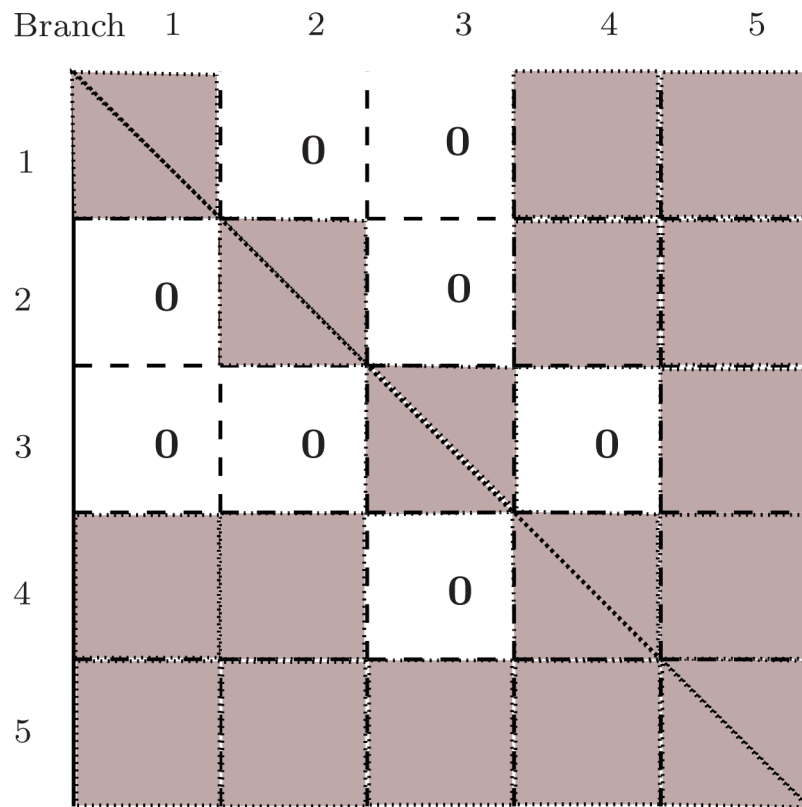


**Fig 4.** Structure of the  $E_A$  spatial operator for the tree topology system in Figure 3

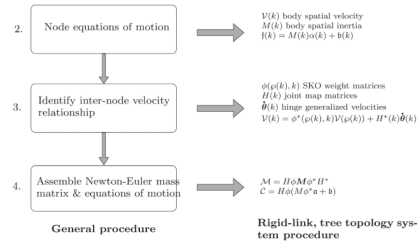




**Fig 5.** Structure of the  $\mathbf{A}$  spatial operator for the tree-topology system in Figure 3



**Fig 6.** Structure of the mass matrix  $\mathcal{M}$  for the tree topology system in Figure 3



**Fig 7.** Flow chart illustrating steps 2 through 4 of the process described in Section 8.1 for developing SKO models for multibody systems. The corresponding steps for rigid-link, tree-topology multibody systems are shown on the right.