



Published in final edited form as:

Multibody Syst Dyn. 2011 June 21; 26(3): 307–333. doi:10.1007/s11044-011-9266-7.

Graph Theoretic Foundations of Multibody Dynamics Part I: Structural Properties

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 is the first part of two papers that use concepts from graph theory to obtain a deeper understanding of the mathematical foundations of multibody dynamics. The key contribution is the development of a unifying framework that shows that key analytical results and computational algorithms in multibody dynamics are a direct consequence of structural properties and require minimal assumptions about the specific nature of the underlying multibody system. This first part focuses on identifying the abstract graph theoretic structural properties of spatial operator techniques in multibody dynamics. The second part paper exploits these structural properties to develop a broad spectrum of analytical results and computational algorithms.

Towards this, we begin with the notion of graph adjacency matrices and generalize it to define block-weighted adjacency (BWA) matrices and their 1-resolvents. Previously developed spatial operators are shown to be special cases of such BWA matrices and their 1-resolvents. These properties are shown to hold broadly for serial and tree topology multibody systems. Specializations of the BWA and 1-resolvent matrices are referred to as spatial kernel operators (SKO) and spatial propagation operators (SPO). These operators and their special properties provide the foundation for the analytical and algorithmic techniques developed in the companion paper.

We also use the graph theory concepts to study the topology induced sparsity structure of these operators and the system mass matrix. Similarity transformations of these operators are also studied. While the detailed development is done for the case of rigid-link multibody systems, the extension of these techniques to a broader class of systems (e.g. deformable links) are illustrated.

1 Introduction

Graph theory provides concepts that are natural for describing multibody systems and their application to robotic and multibody system dynamics is not new. Wittenburg [36] used the underlying graph associated with a multibody system to organize and formulate the equations of motion. Later it was shown that Wittenburg's formulation procedure could in fact be completely re-interpreted using linear graph theory ideas from the alternative vector-network technique [24]. The goal of some of these graph theory methods has been to make systematic the process of generating the complex equations of motion for multibody systems [3, 4]. Some of the extensions have led to computer techniques for organizing and automatically generating symbolic representations of the equations of motion [31]. Graph theory techniques have also been used for the kinematic analysis of mechanisms [37].

Motivated by robotics, control and simulation needs, mathematical operator techniques have been developed by researchers to obtain a system-level understanding of the underlying structure of the system kinematics and dynamics. Researchers [5, 7, 8, 30, 32] have used system-level matrices and operators to analyze and exploit the structure and sparsity of the mass matrix. Mass matrix factorization techniques and system-level global transforms to simplify the coupled equations of motion into diagonalized forms have also been explored [1, 2, 10, 20, 22, 33]. One common feature to most of these techniques has been the use of *relative*, instead of *absolute* coordinates [25] to parametrize the state of the system. While absolute coordinate models are arguably easier to assemble, relative coordinate models use minimal coordinates (for tree-topology systems) and are more suitable from a control perspective. We adopt the relative coordinates approach throughout this paper.

Spatial operator techniques, developed by Rodriguez and others [16, 27–29], have led to the development of important analytical insights and computational dynamics algorithms. The operator techniques have been inspired by mathematical parallels between the structure of the dynamical equations of motion and the time-domain problem of optimal estimation and smoothing. The covariance factorization and optimal filtering techniques behind the seminal Kalman filtering techniques have been shown to be applicable to the dynamics domain. Spatial operators were initially applied to serial-chain rigid body systems to obtain the following analytical factorization and inversion results for serial-chain, rigid body system mass matrices [27, 28]:

$$\begin{aligned}
 \mathcal{M} &= H\phi M\phi^*H^* \\
 \mathcal{M} &= [\mathbf{I}+H\phi\mathcal{K}]\mathcal{D}[\mathbf{I}+H\phi\mathcal{K}]^* \\
 [\mathbf{I}+H\phi\mathcal{K}]^{-1} &= [\mathbf{I}-H\psi\mathcal{K}] \\
 \mathcal{M}^{-1} &= [\mathbf{I}-H\psi\mathcal{K}]^*\mathcal{D}^{-1}[\mathbf{I}-H\psi\mathcal{K}]
 \end{aligned} \tag{1}$$

In the above \mathcal{M} denotes the configuration dependent system mass matrix, and H , ϕ etc. are examples of spatial operators. The results in Eq. 1 have been applied to operational space dynamics [23], under-actuated system dynamics [19], sensitivity analysis [21], dynamics diagonalization [20] etc.

The spatial operator techniques were subsequently extended to dynamics models for systems with rigid and deformable links [18], tree and non-tree topologies [12, 29], geared and flexible hinge systems [17] etc. accompanied by the development of several efficient computational algorithms for robot control and simulation based on these analytical results. While the elements of the H , ϕ and other spatial operators in Eq. 1 varied significantly across these different systems (e.g. rigid versus deformable bodies), remarkably, the mass matrix factorization and inversion relations in Eq. 1 continue to hold across these systems. Despite the convergence of such operator-level mathematical properties, there has been a lack in understanding of the underlying basis for such convergence since each system's spatial operator formulation was developed independently starting from each system's specific properties.

In this paper we identify deep connections between graph theory techniques and multibody system dynamics¹. This paper generalizes the notion of adjacency matrices for graphs to introduce the notion of *block-weighted adjacency (BWA) matrices* and studies their mathematical properties. The primary contribution of this paper is to show that key spatial operators are in fact instances of such BWA matrices for graphs associated with multibody systems. This relationship is quite general and independent of the specific branching

¹Preliminary results in this paper have been reported in [15].

structure of the system topology, the rigid/flexible nature of the component bodies, the body indexing schemes etc. The BWA matrices associated with multibody systems are referred to as *spatial kernel operators (SKO)*, and their related 1-resolvents (defined later) as *spatial propagation operators (SPO)*. We show that the system equations of motions described using these SKO and SPO spatial operators have the same form across a broad family of multibody systems. The companion second part of this paper [14] builds upon this foundation to show that important analytical techniques and computational dynamics algorithms follow directly from these abstract, analytical results. As such, these techniques and algorithms are broadly applicable, with minimal restrictions, to a broad range of multibody systems.

The organization of this paper is as follows. We begin by reviewing the properties of directed graphs (digraphs) in Section 2. We examine the properties of adjacency matrices used to describe graph connectivity in Section 3, and review the nilpotency property of such matrices for trees. Section 4 generalizes the notion of adjacency matrices to *block weighted adjacency (BWA)* matrices with block matrix elements. Section 5 studies the properties of BWA matrices under similarity transformations. In Section 6 we identify the connections between BWA matrices and their 1-resolvents with multibody dynamics by formulating at the dynamics of serial-chain and tree-topology rigid multibody systems. Section 7 shows that these BWA properties extend virtually unchanged to tree-topology systems. It also introduces the SKO and SPO nomenclature. Finally, Section 8 provides a brief overview of how the BWA properties extend beyond rigid body to deformable body multibody systems.

The graph theoretic structural properties developed in this paper provide a unifying perspective for dynamics models across a broad range of multibody systems and lay the groundwork for the examination of analytical and algorithmic properties of multibody systems that are the subject of the companion part II paper [14].

2 Directed graphs and trees

We begin with an overview of terminology and concepts from graph theory. A **graph** is a collection of **nodes**, and **edges** connecting pairs of nodes. A **directed graph** (also known as a **digraph**) is a graph where the edges have direction, i.e., an edge from one node to another is not the same as an edge in the reverse direction [34]. Each edge in a digraph defines a parent/child relationship between the node pair of that edge. Nodes defining an edge are said to be **adjacent** nodes. The node from which the edge emanates is referred to as the **parent node**, and the destination node is said to be the **child node**. The set of parent nodes of the k^{th} node is denoted $\mathcal{P}(k)$, and the set of its children nodes is $\mathcal{C}(k)$. When a node has multiple child nodes, each child node represents the start of a **branch** emanating from the node. Nodes without parent nodes are referred to as **root nodes**. Digraphs can have zero, one, or multiple root nodes. We assume that there is at most a single edge in the same direction between any pair of nodes, i.e., parallel edges between a pair of nodes are not allowed.

A node, j , is said to be the **ancestor** of another node, i , if there is a directed path from the node j to the node i . We use the notation $i < j$ (or equivalently $j > i$) to indicate that node j is an ancestor of node i . The notation $i \not< j$ implies that node j is not an ancestor of node i . Node i is said to be the **descendant** of node j if j is an ancestor of node i . A pair of nodes, i and j , are said to be **related** if one of them is the ancestor of the other; otherwise they are said to be **unrelated**.

Figure 1 shows a taxonomy for graphs and digraphs adapted from Ramoni [26]. We briefly describe the left branch of this classification hierarchy in increasing order of specialization [6, 35]:

1. A **connected digraph** is a digraph such that there is an undirected path connecting any pair of nodes, i.e., it is a digraph without disjoint components.
2. A **rooted digraph** is a connected digraph with a single root node that is the ancestor of every other node in the digraph. All edges connected to the root node are directed away from the root node. Thus, with r denoting the root node, we have $r \succ k$ for all nodes k in a rooted digraph.
3. A **directed acyclic graph (DAG)** is a connected digraph without any directed cycles, i.e., there is no directed path from any node back to itself. Thus, a node cannot be its own ancestor in a DAG, i.e., $k \not\prec k$ for any node k in the digraph. Figure 2a illustrates a DAG.
4. A **simply connected graph**, or a **polytree** (also known as *singly connected networks*), is a DAG in which there is at most one directed path between any pair of nodes. For such digraphs, if nodes i and j are both ancestors of node k , then one of the i and j nodes must be an ancestor of the other. Notationally, this condition states that if $k \prec i$ and $k \prec j$, then either $i \preceq j$ or $j \preceq i$. Removing an edge decomposes a polytree into a pair of polytrees. A node in a polytree can have multiple parent nodes.
5. A **simple tree**, or **tree**, illustrated in Figure 2b, is a polytree where a node has at most one parent node, i.e., $\mathcal{P}(k)$ contains at most one node for any node k . All trees have a unique root node. A tree is a rooted digraph, and is also referred to as a **rooted tree** or an **arborescence**. Removing an edge decomposes a tree into a pair of trees.

While a node in a tree can have at most one parent, there is no restriction on the number of children nodes. The branching structure implies that it is possible for nodes to be on different branches with no directed path connecting them. Thus, trees can have *unrelated* nodes, i.e., nodes that have no directed paths connecting them. Another noteworthy fact is that, for a pair of nodes, i and j , while $j \succ i \implies i \not\prec j$, the converse is not true in general. For a tree, we use the notation $\mathcal{P}(i, j)$ for the closest ancestor node of a pair of nodes i and j .

6. A **serial-chain** is a tree where each node has at most one child. Figure 2e illustrates a serial-chain. Unlike trees, serial-chains have the stronger property that all node pairs are *related*, i.e., for any pair of nodes in the serial-chain, one of the nodes is necessarily an ancestor of the other.
7. An **arborescence forest**, or **forest**, is a collection of disjoint trees. Removing an edge from a tree converts it into a forest. Adding a common root node to the independent trees in a forest converts them into a single tree. Conversely, removing the root node from a tree converts it into a forest.

A tree is said to be **canonical**² if the index of a parent node is always greater than the index of its child node, i.e., $\mathcal{P}(k) > k$ for any node k . The digraph in Figure 2c illustrates a canonical tree. Any tree can be converted into a canonical tree with a suitable renumbering of the nodes. The node numbering for a canonical tree is not unique since the canonical tree requirement imposes only a partial ordering on the node indices. The canonical numbering is, however, unique for a serial-chain.

²Strictly speaking, the canonical property of a tree depends only on the way indices are assigned to the nodes, and not the topological structure of the tree itself.

A **strictly canonical tree** is a canonical tree in which the indices within each of its serial-chain segments are sequential. The digraph in Figure 2d illustrates a strictly canonical tree. The canonical tree in Figure 2c is however not strictly canonical because the (1, 3) and (2, 4) index sets of the nodes in its serial segments are not sequential. Once again, any rooted tree can be converted into a strictly canonical tree with a suitable renumbering of the nodes.

Every rooted digraph has a **spanning tree**, i.e., a tree that contains all the nodes in the digraph and whose edges belong to the digraph. The edges removed to convert a rooted digraph into its spanning tree are referred to as **cut-edges**.

3 Adjacency matrices for digraphs

One way of representing the node/edge connectivity of a digraph is through an **adjacency matrix** for the digraph. For a digraph with n nodes, the adjacency matrix is an $n \times n$ square matrix. The only non-zero elements of an adjacency matrix are for adjacent node pairs. Thus, the $(i, j)^{th}$ element of an adjacency matrix is 1 if and only if the j^{th} node is a child of the i^{th} node, and is 0 otherwise. With the “.” entries indicating 0 entries, adjacency matrices for the digraphs in Figure 2 are shown in Figure 3. **Weighted adjacency matrices** are generalizations of standard adjacency matrices in allowing arbitrary weight entries, instead of just the 1 value for the edges in a digraph.

3.1 Properties of digraph adjacency matrices

The adjacency matrix, \mathbb{S} of a digraph can be expressed as

$$\mathbb{S} = \sum_{j=1}^n \sum_{k \in \mathcal{P}(j)} e_k e_j^* \tag{2}$$

where e_i denotes a vector of length n , containing all zeros except for the i^{th} element which has value 1, i.e.,

$$e_i(r) = \begin{cases} 1 & \text{for } i=r \\ 0 & \text{otherwise} \end{cases} = \mathbb{1}_{[i=r]} \tag{3}$$

$\mathbb{1}_{[<cond>]}$ denotes the **indicator function**, whose value is 1 when the condition specified in its subscript is true, and whose value is 0 otherwise. Observe that

$$e_j^* \cdot e_k = \mathbb{1}_{[j=k]} \tag{4}$$

The number of $e_k e_j^*$ terms in Eq. 2 is precisely the number of edges in the digraph. Each $e_k e_j^*$ edge term is an $n \times n$ matrix, with a single non-zero element of value 1 for the associated (k, j) edge. Key structural properties of an adjacency matrix, \mathbb{S} for a digraph are summarized below:

1. The non-zero entries in the j^{th} column of \mathbb{S} are for the $\mathcal{P}(j)$ parent nodes of the j^{th} node. Thus, the product of \mathbb{S} with the i^{th} unit vector e_i is a vector whose non-zero entries are the parent nodes of the i^{th} node. In particular, a node p is a root node if and only if

$$\mathbb{S} \cdot e_p = \mathbf{0} \tag{5}$$

In other words, root node columns of \mathbb{S} have all zero elements.

2. The non-zero entries in the i^{th} row of \mathbb{S} are for the $C(i)$ child nodes of the i^{th} node. Thus, the product of \mathbb{S} with the i^{th} unit vector, e_i , is a vector whose non-zero entries are the child nodes of the i^{th} node. In particular, a node t is a *tip* node, i.e., it has no children, if and only if

$$e_t^* \cdot \mathbb{S} = \mathbf{0} \quad (6)$$

In other words, tip node rows of \mathbb{S} have all zero elements.

3. Permutation matrices can be used to re-index the nodes in a digraph. Permutation matrices are orthogonal, i.e., the transpose of a permutation matrix is its inverse. With T denoting such a permutation matrix, and \mathbb{S} the adjacency matrix of the original digraph, the adjacency matrix of the digraph after re-indexing is $T\mathbb{S}T^*$.

3.2 Properties of tree adjacency matrices

Since nodes in a tree can have at most one parent node, the adjacency matrix of a tree digraph in Eq. 2 can be expressed in the following simpler form:

$$\mathbb{S} = \sum_{k=1}^n e_{\psi(k)} e_k^* \quad (7)$$

We now examine the properties of an adjacency matrix, \mathbb{S} for a tree digraph.

1. Since a node can have at most one parent in a tree, there can be at most one non-zero entry in any column of \mathbb{S}
2. Only a single column, corresponding to the root node, is all zero in \mathbb{S}
3. A row can have multiple non-zero elements.
4. For *canonical trees*, \mathbb{S} is always strictly *lower-triangular*.
5. Reversing the direction of the edges in a tree typically destroys the tree character; leaving only a polytree. The reversal process converts multiple children nodes into multiple parent nodes. \mathbb{S} is the adjacency matrix for the polytree.
6. \mathbb{S} has simpler structure for *strictly canonical trees*. It consists of the adjacency matrix for serial canonical serial-chain segment digraphs, \mathbb{S}_i , along the diagonal, and off-diagonal $\mathbb{S}_{i,j}$ elements for the edges connecting the serial-chain segments. For the example strictly-canonical tree digraph in Figure 4, \mathbb{S} has the structure shown in Figure 5.
7. The k^{th} power of the adjacency matrix, \mathbb{S}^k , contains non-zero elements only for paths of length k connecting the nodes [9]. That is, the (i, j) element of \mathbb{S}^k is non-zero if and only if there is a directed path of length k from node i to node j .
8. With n denoting the number of nodes in the tree, the n^{th} power of the adjacency matrix, \mathbb{S}^n , is zero. This is because the length of any path in a tree with n nodes is less than n . Thus \mathbb{S} is a nilpotent matrix³. In other words, the adjacency matrix for a tree digraph is always nilpotent.

³A matrix X is said to be nilpotent if there exists an integer, n , such that $X^n = \mathbf{0}$.

3.3 Properties of serial-chain adjacency matrices

We now review the properties of an adjacency matrix, \mathbb{S} for a serial-chain digraph.

1. Since a node can have at most one child in a serial-chain, its adjacency matrix can have at most one non-zero element per row.
2. For a canonical serial-chain, only the first sub-diagonal of \mathbb{S} is non-zero.
3. For a canonical serial-chain, only the k^{th} sub-diagonal of \mathbb{S}^k contains non-zero elements. Each additional power shifts the sub-diagonal one level lower.
4. A reversed serial chain is once again a serial-chain.

4 Block weighted adjacency matrices

Adjacency matrices have 1 or 0 scalar entries for the node pair of each edge in a digraph, while weighted adjacency matrices allow scalar entries other than 1 for the edges. We now generalize weighted adjacency matrices to define **block-weighted adjacency (BWA)** matrices. Instead of scalars, BWA matrices are weighted adjacency matrices whose weight entries for the edges are not scalars, but instead matrices. First, each tree node, j , is assigned a *weight dimension*, m_j , and its parent edge is assigned a *weight matrix* of dimension $m_{\mathcal{P}(j)} \times m_j$, denoted $w(\mathcal{P}(j), j)$. The $(k, j)^{\text{th}}$ block-matrix element of a BWA matrix, \mathbb{S}_W , is thus

$$\mathbb{S}_W(k, j) = \begin{cases} w(k, j) & \text{if } k = \mathcal{P}(j) \\ \mathbf{0} & \text{otherwise} \end{cases} \in \mathcal{R}^{m_k \times m_j} \quad (8)$$

Other than the dimensionality requirement, there are no additional restrictions on the $\mathbb{S}_W(k, j)$ weight matrices. The standard adjacency matrix, \mathbb{S} in Eq. 2 is a special case of a BWA matrix, where all the node dimensions are 1, and the edge weights are the scalar 1. \mathbb{S}_W has non-zero block-entries only for node pairs corresponding to edges in the digraph. It is easy to verify from the definition of the block-elements in Eq. 8 that \mathbb{S}_W is a well-defined *square matrix* of dimension

$$N \triangleq \sum_{k=1}^n m_k \quad (9)$$

We extend the notion of the e_k unit vectors in Eq. 3, to the BWA context by assigning them block matrix entries as follows:

$$e_k(i) = \begin{cases} \mathbf{I}_{m_k} & \text{if } i = k \\ \mathbf{0}_{m_i \times m_k} & \text{otherwise} \end{cases} \quad (10)$$

Thus, the e_k unit “vector” is a matrix of dimension $N \times m_k$, with n block matrix elements. The new version of Eq. 4 is

$$e_j^* \cdot e_k = \begin{cases} \mathbf{I}_{m_j} & \text{if } j = k \\ \mathbf{0}_{m_j \times m_k} & \text{otherwise} \end{cases} \quad (11)$$

Using the unit vectors, and the component definitions in Eq. 8, \mathbb{S}_W can be expressed as:

$$\mathbb{S}_w = \sum_{j=1}^n \sum_{k \in \varphi(j)} e_k w(k, j) e_j^* \tag{12}$$

Eq. 12 is a generalization of Eq. 2 for BWA matrices.

4.1 BWA matrices for tree digraphs

Analogous to Eq. 7, the BWA matrix expression in Eq. 12 has the following simpler form for tree digraphs:

$$\mathbb{S}_w = \sum_{k=1}^n e_{\varphi(k)} w(\varphi(k), k) e_k^* \tag{13}$$

The following lemma provides an expression for the block elements of \mathbb{S}_w^k for tree digraphs.

Lemma 1 Elements of \mathbb{S}_w^k for a tree digraph

For a tree digraph, \mathbb{S}_w^k , the k^{th} power of the \mathbb{S}_w BWA matrix, contains non-zero block-elements only for node pairs connected by directed paths of length k . The value of its block-element is the product of the k weights for the edges along the path, i.e., the non-zero elements are of the form

$$\mathbb{S}_w^k (\varphi^k(i), i) = w(\varphi^k(i), \varphi^{k-1}(i)) * \dots * w(\varphi(i), i) \in \mathcal{R}_{\varphi^k(i) \times i}^{m \times m_i} \tag{14}$$

Here, $\varphi^k(i)$ denotes the k^{th} ancestor of the i^{th} node.

Proof Let us illustrate the proof for $k = 2$. \mathbb{S}_w^2 is given by

$$\begin{aligned} \mathbb{S}_w^2 &\stackrel{13}{=} \left(\sum_{j=1}^n e_{\varphi(j)} w(\varphi(j), j) e_j^* \right) * \left(\sum_{i=1}^n e_{\varphi(i)} w(\varphi(i), i) e_i^* \right) \\ &\stackrel{4}{=} \sum_{j=1}^n \sum_{i=1}^n e_{\varphi(j)} w(\varphi(j), j) \mathbb{1}_{[j=\varphi(i)]} w(\varphi(i), i) e_i^* \\ &= \sum_{i=1}^n e_{\varphi^2(i)} w(\varphi^2(i), \varphi(i)) * w(\varphi(i), i) e_i^* \end{aligned}$$

The $(\varphi^2(i), i)$ block-element of \mathbb{S}_w^2 is thus,

$$\mathbb{S}_w^2 (\varphi^2(i), i) = w(\varphi^2(i), \varphi(i)) * w(\varphi(i), i)$$

which establishes Eq. 14 for \mathbb{S}_w^2 . Continuing in a similar vein, Eq. 14 can be established for arbitrary k .

4.2 The 1-resolvent of tree BWA matrices

The following lemma establishes the nilpotency of \mathbb{S}_W for tree digraphs.

Lemma 2 Nilpotency of tree BWA matrices

The n^{th} power of a tree BWA matrix, \mathbb{S}_W^n , is zero, where n denotes the number of nodes in the system. Hence, the BWA matrix for a tree digraph is a nilpotent matrix.

Proof Lemma 1 states that the non-zero entries of \mathbb{S}_W^n correspond to node pairs connected by paths of length n . Since paths can be of length at most $n - 1$ in a tree with n nodes, it follows that \mathbb{S}_W^n is the zero matrix.

In matrix theory, the resolvent of a matrix A , is defined as the $(\lambda \mathbf{I} - A)^{-1}$ matrix for a scalar λ . We use the term, **1-resolvent**, to denote the specific resolvent with $\lambda = 1$.

Lemma 3 The 1-resolvent of a nilpotent matrix

If U is a nilpotent matrix, such that $U^n = \mathbf{0}$, then its 1-resolvent, $W \triangleq (\mathbf{I} - U)^{-1}$, is given by

$$W = \mathbf{I} + U + U^2 + \dots + U^{n-1} \quad (15)$$

Proof For W as defined in Eq. 15

$$UW = WU = U + U^2 + \dots + U^n = U + U^2 + \dots + U^{n-1} = W - \mathbf{I}$$

Rearranging terms, we have

$$\mathbf{I} = W - UW = (\mathbf{I} - U)W \Rightarrow (\mathbf{I} - U)^{-1} = W$$

The following lemma derives an explicit expression for the 1-resolvent of a tree BWA matrix.

Lemma 4 The 1-resolvent of a tree BWA matrix

The 1-resolvent, $(\mathbf{I} - \mathbb{S}_W)^{-1}$, of a tree BWA matrix exists and can be expressed as the following disjoint sum of its powers:

$$(\mathbf{I} - \mathbb{S}_W)^{-1} = \mathbf{I} + \mathbb{S}_W + \mathbb{S}_W^2 + \dots + \mathbb{S}_W^{n-1} \quad (16)$$

Proof Lemma 2 states that the adjacency matrix of a tree, \mathbb{S}_W , is nilpotent. Eq. 16 follows by applying the expression from Lemma 3 for the 1-resolvent of a nilpotent matrix to \mathbb{S}_W .

Lemma 4 is an important result. It provides an explicit expression for the 1-resolvent of the \mathbb{S}_W BWA matrix for tree systems. For cyclic digraphs, i.e., ones with *directed loops*, the 1-resolvent is not defined because \mathbb{S}_W is not nilpotent and $(\mathbf{I} - \mathbb{S}_W)$ is non-invertible. This topological difference has significant implications for the dynamics properties of tree versus cyclic digraph multibody systems. This is discussed in more detail in [14]

We now introduce notation that better reflects the intimate relationship between a BWA matrix and its 1-resolvent matrix for a tree digraph. With \mathbb{A} denoting the 1-resolvent matrix

for a tree, we will use the notation \mathcal{E}_A to denote its associated BWA matrix. Thus, the relationship between the earlier notation and the new notation is as follows:

$$A \equiv (I - S_w)^{-1} \text{ and } \mathcal{E}_A \equiv S_w = I - A^{-1} \tag{17}$$

The following lemma describes the expression for the block elements of a 1-resolvent matrix.

Lemma 5 Elements of a 1-resolvent matrix A

The (k, j) element of the 1-resolvent matrix of a tree, $A = (I - \mathcal{E}_A)^{-1}$, is given by

$$A(i, j) = \begin{cases} \mathbf{0}_{m_i \times m_j} & \text{for } i \neq j \\ I_{m_i} & \text{for } i=j \\ w(i, j) & \text{for } i=\varphi(j) \\ w(\varphi^k(j), \varphi^{k-1}(j)) * \dots * w(\varphi(j), j) & \text{for } i=\varphi^k(j) \end{cases} \tag{18}$$

Proof Eq. 18 follows from the expression for $(I - \mathcal{E}_A)^{-1}$ in Eq. 16, together with Lemma 1 which describes the elements of the powers of \mathcal{E}_A .

Thus, A has identity matrices along the diagonal. The non-zero block elements correspond to (i, j) node pairs where node i is an ancestor of node j . For such cases, the block element is the product of the weight matrices associated with each of the edges in the path from i to j . From Eq. 18, it is clear that the A -resolvent is sparse, with its sparsity structure directly reflecting the topological structure of the associated tree.

Remark 4.1 Structure of canonical serial-chain \mathcal{E}_A and A matrices

Let us now examine the structure of the 1-resolvent matrix A for a canonical serial-chain digraph. We assume for this example that the serial-chain has 4 nodes, and thus $\mathcal{E}_A^4 = \mathbf{0}$. Using Eq. 16, A is given by the following disjoint sum:

$$A = I + \mathcal{E}_A + \mathcal{E}_A^2 + \mathcal{E}_A^3$$

The non-zero elements of the above matrices are as shown below (with “.” entries indicating zero blocks and the X entries the non-zero ones):

$$\mathcal{E}_A = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \cdot \\ \cdot & \cdot & X & \cdot \end{pmatrix}, \mathcal{E}_A^2 = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \cdot \end{pmatrix}, \mathcal{E}_A^3 = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \cdot & \cdot \end{pmatrix}$$

\mathcal{E}_A has non-zero block entries only along its first sub-diagonal. Each subsequent power of \mathcal{E}_A shifts the non-zero sub-diagonal lower, until it vanishes entirely for the n^{th} power.

Thus, A has the following structure:

$$\mathbb{A} = \begin{pmatrix} \mathbf{I} & \cdot & \cdot & \cdot \\ X & \mathbf{I} & \cdot & \cdot \\ X & X & \mathbf{I} & \cdot \\ X & X & X & \mathbf{I} \end{pmatrix}$$

\mathbb{A} is lower-triangular, with all of the diagonal and lower-triangular block-elements being non-zero. All of these terms are non-zero because all nodes in serial-chains are related, i.e., there is a directed path between all node pairs.

This example shows that, for a canonical serial-chain, all of the lower-triangular elements of \mathbb{A} are non-zero. The following example looks at the structure of a 1-resolvent matrix for a canonical tree.

Remark 4.2 Structure of canonical tree $\mathcal{E}_{\mathbb{A}}$ and \mathbb{A} matrices

The sparsity structure of the $\mathcal{E}_{\mathbb{A}}$ and \mathbb{A} matrices for the canonical tree in Figure 2c is shown below. The “ \cdot ” entries indicate zero block-elements.

$$\mathcal{E}_{\mathbb{A}} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & X & X & X & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & X \end{pmatrix} \text{ and } \mathbb{A} = \begin{pmatrix} \mathbf{I} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \mathbf{I} & \cdot & \cdot & \cdot & \cdot & \cdot \\ X & \cdot & \mathbf{I} & \cdot & \cdot & \cdot & \cdot \\ \cdot & X & \cdot & \mathbf{I} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \mathbf{I} & \cdot & \cdot \\ X & X & X & X & X & \mathbf{I} & \cdot \\ X & X & X & X & X & X & \mathbf{I} \end{pmatrix}$$

These matrices illustrate that, for canonical trees, the $\mathcal{E}_{\mathbb{A}}$ matrix is strictly lower-triangular, and \mathbb{A} is lower-triangular. However, some of the lower-triangular blocks of \mathbb{A} are zero, in contrast with the all non-zero case for canonical serial-chains. The zero terms correspond to the unrelated nodes in the tree, i.e., node pairs that are not connected by a path. Jain [14] takes a more detailed look at the sparsity structures of the $\mathcal{E}_{\mathbb{A}}$ and \mathbb{A} matrices.

This example shows that there is significant sparsity in the lower-triangular blocks of \mathbb{A} for a tree. The sparsity is a function of the branching topological structure of the tree. Serial-chain systems represent the most dense topological structure since all of their nodes are related. Trees, on the other hand, can have unrelated nodes, i.e., nodes with no directed paths connecting them. This decreased connectivity of the nodes is reflected in the sparsity of \mathbb{A} .

The following lemma highlights the semi-group properties of the block elements of \mathbb{A} .

Lemma 6 Semi-group property of $\mathbb{A}(i, j)$ elements

For a tree, let $i, j,$ and k denote nodes, where the i^{th} node is an ancestor of the j^{th} node, and the k^{th} node is on the path connecting them. Then, the block-elements of the \mathbb{A} -resolvent satisfy the following relationship:

$$\mathbb{A}(i, j) = \mathbb{A}(i, k) \mathbb{A}(k, j) \forall k: i \geq k \geq j \tag{19}$$

This property is also known as the **semi-group** property for the elements of a 1-resolvent matrix.

Proof Let $i = \mathcal{P}^m(j)$ for some m , and $k = \mathcal{P}^r(j)$ for some $r < m$. Let l denote the child node of k that is on the path to node j . From the Eq. 18 expression for the block elements of \mathbb{A} we have

$$\begin{aligned} \mathbb{A}(i, j) & \stackrel{18}{=} w(\varphi^m(j), \varphi^{m-1}(j)) * \dots * w(\varphi(j), j) \\ & = [w(\varphi^m(j), \varphi^{m-1}(j)) * \dots * w(\varphi(k), k)] * [w(k, l) * \dots * w(\varphi(j), j)] \\ & \stackrel{18}{=} \mathbb{A}(\varphi^m(j), k) * \mathbb{A}(k, j) = \mathbb{A}(i, k) * \mathbb{A}(k, j) \end{aligned}$$

Define the $\tilde{\mathbb{A}}$ matrix derived from the \mathbb{A} -resolvent matrix as follows:

$$\tilde{\mathbb{A}} \triangleq \mathbb{A} - \mathbf{I} \quad (20)$$

$\tilde{\mathbb{A}}$ is strictly lower-triangular for canonical trees. The following lemma establishes some basic properties of $\tilde{\mathbb{A}}$.

Lemma 7 The $\tilde{\mathbb{A}}$ spatial operator

For a 1-resolvent \mathbb{A} the following identities hold:

$$\tilde{\mathbb{A}} = \mathbb{A} - \mathbf{I} = \mathcal{E}_{\mathbb{A}} \mathbb{A} = \mathbb{A} \mathcal{E}_{\mathbb{A}} \quad (21)$$

Proof For any matrix X , such that $(\mathbf{I} - X)$ is invertible, the following matrix identity holds:

$$X(\mathbf{I} - X)^{-1} = (\mathbf{I} - X)^{-1}X = (\mathbf{I} - X)^{-1} - \mathbf{I}$$

With $X = \mathcal{E}_{\mathbb{A}}$, the above equation, together with Eq. 20, directly lead to Eq. 21.

The following exercise extends the adjacency matrix properties in Eq. 5 and Eq. 6 to BWA matrices and their 1-resolvents.

Corollary 4.1 Root/tip nodes and BWA matrices

1. A node r is a root node of the tree if and only if

$$\mathcal{E}_{\mathbb{A}} \cdot e_r = \mathbf{0}, \mathbb{A} \cdot e_r = e_r \quad (22)$$

2. A node t is a tip node of the tree, i.e., it has no children nodes, if and only if

$$\mathcal{E}_{\mathbb{A}}^* \cdot e_t = \mathbf{0}, \mathbb{A}^* \cdot e_t = e_t \quad (23)$$

Proof:

1. The first part of Eq. 22 is a generalized restatement of Eq. 5 for BWA matrices. The latter half follows from the following:

$$\mathbb{A}^{-1} \cdot e_r \stackrel{17}{=} (\mathbf{I} - \mathcal{E}_{\mathbb{A}}) e_r \stackrel{22}{=} e_r$$

2. The proof here is completely analogous to that of the first part with the use of Eq. 6.

The following remark describes properties of the product of a general matrix, and a diagonal matrix, that we will use in the forthcoming sections.

Remark 4.3 Matrix products

Recall that any block-matrix, Y , can be expressed as

$$Y = \sum_{j=1}^n \sum_{k=1}^n e_j Y(j, k) e_k^*$$

Let $X \triangleq \text{diag} \{X(i)\}_{i=1}^n$ and $Z \triangleq \text{diag} \{Z(i)\}_{i=1}^n$ denote compatible block-diagonal matrices. Then it is easy to verify that

$$\begin{aligned} XY &= \sum_{j=1}^n \sum_{k=1}^n e_j X(j) Y(j, k) e_k^* \\ \text{and } YZ &= \sum_{j=1}^n \sum_{k=1}^n e_j Y(j, k) Z(k) e_k^* \end{aligned} \tag{24}$$

While the discussion in this section has been devoted to BWA matrices and 1-resolvents for tree digraphs, all of the properties and results extend directly to forests, i.e., to digraphs consisting of a set of disjoint trees.

5 Similarity transformations of a tree BWA matrix

We now examine the effect of applying similarity transformations to a BWA matrix.

Lemma 8 Similarity transformation of a BWA matrix

Let T denote an invertible matrix and $\mathcal{E}_{\mathbb{A}}$ a tree BWA matrix. Let $Y \triangleq T \mathcal{E}_{\mathbb{A}} T^{-1}$ denote a similarity transformation of $\mathcal{E}_{\mathbb{A}}$ by the T matrix. Then Y is nilpotent, and its 1-resolvent is given by the following similarity transformation:

$$(I - Y)^{-1} = T \mathbb{A} T^{-1} \tag{25}$$

Proof Since $\mathcal{E}_{\mathbb{A}}$ is nilpotent, we have

$$Y^n = (T \mathcal{E}_{\mathbb{A}} T^{-1})^n = T (\mathcal{E}_{\mathbb{A}})^n T^{-1} = \mathbf{0}$$

Hence, Y is nilpotent. Also,

$$(I - Y)^{-1} = (I - T \mathcal{E}_{\mathbb{A}} T^{-1})^{-1} = T (I - \mathcal{E}_{\mathbb{A}})^{-1} T^{-1} = T \mathbb{A} T^{-1}$$

This establishes Eq. 25.

The above lemma provides an expression for the 1-resolvent of the matrix obtained by applying a similarity transformation to a BWA matrix. However, there is no guarantee that the transformed matrix, Y , retains the block partitioned structure of the original BWA matrix. Generally, the transformed Y matrix will not be a BWA matrix for the tree.

5.1 Permutation similarity transformations

A special case, where the BWA property is preserved under similarity transformations, occurs when the transformation matrix is a **permutation matrix** used to re-index the nodes in the tree. For such a permutation matrix, denoted T , we know that $T^{-1} = T^*$. The transformed matrix, $\mathcal{E}_{\mathbb{B}} \triangleq T \mathcal{E}_{\mathbb{A}} T^{-1}$ is indeed a BWA matrix for the tree, and its 1-resolvent \mathbb{B} is given by:

$$\mathcal{E}_{\mathbb{B}} = T \mathcal{E}_{\mathbb{A}} T^* \quad \text{and} \quad \mathbb{B} \triangleq (\mathbf{I} - \mathcal{E}_{\mathbb{B}})^{-1} \underline{\underline{25}} T \mathbb{A} T^* \quad (26)$$

In the multibody context, such permutation transformations relate equations of motion obtained using different body numbering schemes. In particular, any tree can be converted into a canonical tree with an appropriate re-labeling of the nodes. This implies that the $\mathcal{E}_{\mathbb{A}}$ and \mathbb{A} matrices for any tree are related to the lower-triangular versions for the corresponding canonical tree via permutation matrices.

5.2 Similarity-shift transformations

Another type of similarity transformation, referred to as a **similarity-shift transformation**, that preserves the BWA property is discussed next. Let $\Delta_{\mathbb{C}}$ be an invertible, block-diagonal matrix defined as follows:

$$\Delta_{\mathbb{C}} \triangleq \text{diag} \{ \Delta_{\mathbb{C}}(k) \}_{k=1}^n \quad \text{where} \quad \Delta_{\mathbb{C}}(k) \in \mathcal{R}^{m_k \times m_k} \quad (27)$$

Let $\mathbb{B}(\varphi(k), k)$ denote new weight matrices defined from the original $\mathbb{A}(\varphi(k), k)$ weight matrices as follows:

$$\mathbb{B}(\varphi(k), k) = \Delta_{\mathbb{C}}(\varphi(k)) \mathbb{A}(\varphi(k), k) \Delta_{\mathbb{C}}^{-1}(k) \quad (28)$$

Let $\mathcal{E}_{\mathbb{B}}$ denote the BWA matrix with the $\mathbb{B}(\varphi(k), k)$ weight matrices. The following lemma establishes the relationship between the $\mathcal{E}_{\mathbb{A}}$ and $\mathcal{E}_{\mathbb{B}}$ BWA matrices and their 1-resolvents.

Lemma 9 Similarity-shift transformation of a BWA matrix

Let $\mathcal{E}_{\mathbb{A}}$ and $\mathcal{E}_{\mathbb{B}}$ be BWA matrices whose weight matrices satisfy Eq. 28. Then the BWA matrices $\mathcal{E}_{\mathbb{B}}$ and $\mathcal{E}_{\mathbb{A}}$ are related by the following similarity transformation:

$$\mathcal{E}_{\mathbb{B}} = \Delta_{\mathbb{C}} \mathcal{E}_{\mathbb{A}} \Delta_{\mathbb{C}}^{-1} \quad (29)$$

Also, the 1-resolvent matrices, $\mathbb{A} = (\mathbf{I} - \mathcal{E}_{\mathbb{A}})^{-1}$ and $\mathbb{B} = (\mathbf{I} - \mathcal{E}_{\mathbb{B}})^{-1}$, are related by the same similarity transformation:

$$\mathbb{B} = \Delta_{\mathbb{C}} \mathbb{A} \Delta_{\mathbb{C}}^{-1} \quad (30)$$

Proof Now

$$\begin{aligned}
 \mathcal{E}_B &\stackrel{28}{=} \sum_{k=1}^n e_{\varphi(k)} \Delta_C(\varphi(k)) \mathbb{A}(\varphi(k), k) \Delta_C^{-1}(k) e_k^* \\
 &\stackrel{24}{=} \sum_{k=1}^n \Delta_C e_{\varphi(k)} \mathbb{A}(\varphi(k), k) e_k^* \Delta_C^{-1} \\
 &= \Delta_C \sum_{k=1}^n e_{\varphi(k)} \mathbb{A}(\varphi(k), k) e_k^* \Delta_C^{-1} = \Delta_C \mathcal{E}_A \Delta_C^{-1}
 \end{aligned}$$

This establishes Eq. 29. Further application of Lemma 8 leads to Eq. 30.

In the multibody context, similarity-shift transformations are used to define the relationship between equations of motion associated with different choices of body frames.

6 Multibody System Graphs

Digraphs provide natural mathematical constructs for describing the topology and connectivity of bodies in a multibody system. They have been used for the systematic formulation of the equations of motion [24, 31, 36], as well as for kinematical analysis [37].

We define the **standard** digraph associated with a multibody system as one with the inertial frame as the root node, and all the links in the system as the remaining nodes in the digraph. Thus, an n -link multibody system has a digraph with $n+1$ nodes.

The edges in the digraph are defined by the motion constraints among the bodies, and between the bodies and the inertial frame. Thus, each hinge is represented by an edge, with the edges orientated from the inboard to the outboard body. Additional edges are assigned to other non-hinge motion constraints in the system. All motion constraints with respect to the inertial frame are defined so that edges from the inertial frame node to the link nodes are directed away from the inertial frame root node. These assignments result in a rooted digraph representation for the multibody system. Figure 6 illustrates the correspondence between the bodies and hinges in a serial-chain multibody system and the standard, serial-chain digraph for the system. The convention is to depict the inertial frame node as unfilled, and the edges from the node as dashed lines. Multibody systems are classified as follows, based on the topology of Serial-chain multibody system Serial-chain digraph their standard digraph:

- systems with tree standard digraphs are referred to as **tree-topology systems**.
- systems with serial-chain standard digraphs are referred to as the familiar **serial-chain systems**. They are special cases of tree-topology systems.
- systems with non-tree standard digraphs are referred to as **closed-chain** or **constrained systems**. These digraphs can have directed cycles and/or multiply-connected nodes. Recall that every rooted digraph can be decomposed (non-uniquely) into a spanning tree together with a set of cut-edges. A decomposition into a spanning tree with $n + 1$ nodes and a set of cut-edges is often used when working with closed-chain systems.

Removing the inertial root node converts the spanning tree into a forest. We will refer to this forest as the **SKO-forest** for the system⁴. The SKO-forest has just n nodes, matching the number of bodies in the system. Nodes corresponding to base-bodies in the multibody

⁴The SKO terminology is defined later in Section 7.1.1.

system are root nodes in the SKO-forest. We will make extensive use of n -node SKO-forests since they allow us to work with adjacency matrices of dimension n instead of dimension $n + 1$. There is no information loss because the spanning tree can be fully recovered by adding the inertial frame root node and its edges to the root nodes in the SKO-forest. The inertial frame node is, in reality, redundant in the spanning tree of a multibody system since it is always present as the root node.

Figure 7 illustrates several examples of multibody systems and their associated standard digraph representations, as described below:

1. A serial-chain system with a serial-chain digraph.
2. A tree-topology system consisting of a pair of independent serial manipulators. The resulting digraph is a tree.
3. A tree-topology system with branching reflected in its tree digraph.
4. A closed-chain four-bar linkage with an internal loop constraint. Based on the directionality of the constraint hinge, the digraph contains a directed cycle or is multiply-connected.
5. A closed-chain pair of serial-manipulators whose end-effectors are constrained. The digraph for this system is a multiply-connected.
6. A closed-chain serial manipulator with constrained end-effector. The digraph is multiply-connected.

Non-standard digraphs where the bodies and digraph nodes are not in one-to-one correspondence with each other are permitted - and are used in constraint embedding techniques for modeling closed-chain dynamics [13]

6.1 BWA Matrices and Serial-Chain Rigid Body Systems

In this section we consider serial-chain multibody systems to begin establishing connections between BWA matrices and their 1-resolvents with multibody system dynamics.

6.1.1 Review of serial-chain velocity kinematics—Consider an n -link rigid body serial-chain system. The tip link is denoted link 1 and the base-body link as link n . The associated serial-chain tree associated with this graph is a strictly canonical tree with the parent/child relationship given by $\mathcal{P}(k) = k + 1$.

Using spatial coordinate-free notation with $\mathcal{V}(k)$ denoting the spatial velocity (angular +linear, 6-dimensional), $\dot{\theta}(k)$ the hinge generalized velocities, $H^*(k)$ the hinge map matrix, and

$$\phi^*(k+1, k) = \begin{pmatrix} \mathbf{I} & \tilde{l}(k+1, k) \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (31)$$

the rigid body transformation matrix, the link-to-link spatial velocity relationships can be expressed as [7, 11]:

$$\mathcal{V}(k) = \phi^*(k+1, k) \mathcal{V}(k+1) + H^*(k) \dot{\theta}(k) \quad (32)$$

The overall velocity degrees of freedom for the system is denoted \mathcal{N} and is defined as the sum of all the individual hinge degrees of freedom. Now we introduce **stacked** vectors needed to define system level relationships. The stacked vectors \mathcal{V} and θ are defined as

$$\mathcal{V} \triangleq \text{col} \{ \mathcal{V}(k) \}_{k=1}^n = \begin{bmatrix} \mathcal{V}(1) \\ \mathcal{V}(2) \\ \vdots \\ \mathcal{V}(n) \end{bmatrix} \in \mathcal{R}^{6n}, \text{ and } \theta \triangleq \text{col} \{ \theta(k) \}_{k=1}^n = \begin{bmatrix} \theta(1) \\ \theta(2) \\ \vdots \\ \theta(n) \end{bmatrix} \in \mathcal{R}^{\mathcal{N}} \quad (33)$$

The \mathcal{V} stacked vector consists of the component body-level $\mathcal{V}(k)$ spatial velocity vectors combined into a single large vector. Correspondingly the θ stacked vector consists of the component body-level $\theta(k)$ generalized coordinates combined into a single large vector. The link-level Eq. 32 relationship can now be expressed equivalently at the system level [11] as

$$\mathcal{V} = \mathcal{E}_\phi^* \mathcal{V} + H^* \dot{\theta} \quad (34)$$

where the spatial operator \mathcal{E}_ϕ is defined as

$$\mathcal{E}_\phi = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \phi(2, 1) & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \phi(3, 2) & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \phi(n, n-1) & \mathbf{0} \end{pmatrix} \in \mathcal{R}^{6n \times 6n} \quad (35)$$

and the block-diagonal $H \in \mathcal{R}^{\mathcal{N} \times 6n}$ spatial operator as

$$H \triangleq \text{diag} \{ H(k) \}_{k=1}^n = \begin{pmatrix} H(1) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & H(2) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & H(n) \end{pmatrix} \in \mathcal{R}^{\mathcal{N} \times 6n} \quad (36)$$

Observe that Eq. 34 is an implicit equation with \mathcal{V} appearing on the both sides. However, it is explicit at the component-level, i.e. if we look at the k^{th} row of this matrix equation, we obtain back the Eq. 32 explicit form.

6.1.2 \mathcal{E}_ϕ is a BWA matrix—We can re-express Eq. 35 as

$$\mathcal{E}_\phi = \sum_{k=1}^n e_{k+1} \phi(k+1, k) e_k^* = \sum_{k=1}^n e_{\varphi(k)} \phi(\varphi(k), k) e_k^* \quad (37)$$

We see that \mathcal{E}_ϕ in Eq. 37 has the structure of a BWA matrix in Eq. 13. Indeed, \mathcal{E}_ϕ is a BWA matrix for the canonical serial-chain tree with 6×6 weight matrices defined by

$$w(\varphi(k), k) = w(k+1, k) \triangleq \phi(k+1, k) = \phi(\varphi(k), k) \quad (38)$$

Since \mathcal{E}_ϕ is a BWA matrix, by Lemma 4, its 1-resolvent $\phi = (I - \mathcal{E}_\phi)^{-1}$ is well-defined, and is given by

$$\phi = \begin{pmatrix} I & 0 & \dots & 0 \\ \phi(2, 1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n, 1) & \phi(n, 2) & \dots & I \end{pmatrix} \in \mathcal{R}^{6n \times 6n} \quad (39)$$

The implicit relationship in Eq. 34 can now be converted into an explicit relationship for \mathcal{V} by rearranging it into the form

$$(I - \mathcal{E}_\phi^*)\mathcal{V} = H^*\dot{\theta} \Rightarrow \phi^{-*}\mathcal{V} = H^*\dot{\theta} \Rightarrow \mathcal{V} = \phi^*H^*\dot{\theta} \quad (40)$$

Thus we have established that the \mathcal{E}_ϕ and ϕ spatial operators for serial-chain canonical serial-chain multibody systems are in fact BWA matrices and 1-resolvents respectively for the associated serial-chain graph. Indeed the strictly lower-triangular and lower-triangular structures of \mathcal{E}_ϕ and ϕ are consistent with their being associated with canonical serial-chain trees.

6.1.3 Equations of motion for serial-chain systems—From the free-body equations of motion of the k^{th} link we have:

$$\ddot{\mathbf{f}}(k) - \phi(k, k-1)\ddot{\mathbf{f}}(k-1) = M(k)\alpha(k) + \mathbf{b}(k) \quad (41)$$

where $\mathbf{f}(k)$ denotes the inter-link spatial force across the k^{th} hinge, $M(k)$ denotes the spatial inertia of the k^{th} link, $\alpha(k)$ its spatial acceleration and $\mathbf{b}(k)$ the gyroscopic spatial force for the k^{th} link [11]. Rearranging the above we obtain the following:

$$\ddot{\mathbf{f}}(k) = \phi(k, k-1)\ddot{\mathbf{f}}(k-1) + M(k)\alpha(k) + \mathbf{b}(k) \quad (42)$$

Switching to the system-level stacked vector re-expression of Eq. 42, leads to the following relationship:

$$\ddot{\mathbf{f}} = \mathcal{E}_\phi^* \ddot{\mathbf{f}} + M\alpha + \mathbf{b} \quad (43)$$

In the above \mathbf{f} , α and \mathbf{b} are stacked vectors along the lines of Eq. 33, α is a stacked vector of body spatial accelerations, while M is a block-diagonal spatial operator with the $M(k)$ elements along the diagonal. Using Eq. 40 and Eq. 43 it is easy to establish that the overall equations of motion are given by:

$$\begin{aligned} \mathcal{V} &= \mathcal{E}_\phi^* \mathcal{V} + H^* \dot{\theta} \\ \alpha &= \mathcal{E}_\phi^* \alpha + H^* \dot{\theta} + \mathbf{a} \\ \ddot{\mathbf{f}} &= \mathcal{E}_\phi^* \ddot{\mathbf{f}} + M\alpha + \mathbf{b} \\ \mathcal{F} &= H\ddot{\mathbf{f}} \end{aligned} \quad (44)$$

In the above \mathcal{F} is the stacked vector of all the hinge generalized forces, while \mathbf{a} is the stacked vector of the Coriolis spatial accelerations for all the bodies. Using the 1-resolvent, ϕ , these

implicit expressions can be transformed from implicit ones to the following explicit version of the equations of motion:

$$\begin{aligned}\mathcal{V} &= \phi^* H^* \dot{\theta} \\ \alpha &= \phi^* (H^* \dot{\theta} + a) \\ \mathfrak{f} &= \phi(M\alpha + b) \\ \mathcal{T} &= H\mathfrak{f}\end{aligned}\quad (45)$$

Combining the expressions in Eq. 45 we obtain

$$\mathcal{T} = \mathcal{M}\dot{\theta} + \mathcal{C} \quad (46)$$

where

$$\mathcal{M} = H\phi M \phi^* H^* \in \mathcal{R}^{\mathcal{N} \times \mathcal{N}} \quad \text{and} \quad \mathcal{C} \triangleq H\phi(M\phi^* a + b) \in \mathcal{R}^{\mathcal{N}} \quad (47)$$

$\mathcal{M} \in \mathcal{R}^{\mathcal{N} \times \mathcal{N}}$ denotes the mass matrix for the serial-chain system, and $\mathcal{C} \in \mathcal{R}^{\mathcal{N}}$ is the vector of velocity dependent nonlinear Coriolis and velocity dependent terms. The expression for the mass matrix in Eq. 47 is referred to as the Newton-Euler Factorization of the mass matrix [11] and was also shown earlier in Eq. 1. This is our first encounter of the convergence of the new concepts of BWA and 1-resolvent matrices with those of the \mathcal{E}_ϕ and ϕ spatial operators that play a key role in spatial operator based dynamics formulations [11].

6.2 Non-canonical serial-chains

For non-canonical serial-chains, the $(k+1)^{th}$ link is not necessarily the parent body of the k^{th} link. So a natural question is whether the \mathcal{E}_ϕ and ϕ spatial operators for non-canonical serial-chains continue to be BWA and 1-resolvent matrices? The answer is in the affirmative. We simply need to discard the implicit assumption that the parent body of the k^{th} link is the $(k+1)^{th}$ link, and switch to the more general $\mathcal{P}(k)$ notation for the parent link. The generalization of Eq. 32 for non-canonical serial-chain systems is:

$$\mathcal{V}(k) = \phi^*(\mathcal{P}(k), k) \mathcal{V}(\mathcal{P}(k)) + H^*(k) \dot{\theta}(k) \quad (48)$$

The velocity relationship in Eq. 34 continues to hold, with \mathcal{E}_ϕ defined by the more general second half of Eq. 37. Thus, \mathcal{E}_ϕ is once again a BWA matrix. Indeed, the rest of the development in Section 6.1.3, leading to the equations of motion in Eq. 44 and Eq. 47, continues to apply. However, since the system is non-canonical, neither \mathcal{E}_ϕ nor ϕ in Eq. 37 and Eq. 39, respectively, are lower-triangular. Thus, even though the component-level equations and the operator structure depend on the body indexing scheme, the BWA and 1-resolvent properties of the \mathcal{E}_ϕ and ϕ spatial operators, and the operator forms of the equations of motion, remain intact! In the next section, we will encounter this theme again for tree-topology systems.

7 BWA matrices and tree-topology, rigid body systems

The key difference between serial-chain and tree-topology multibody systems is that, in a tree system, bodies can have multiple children bodies. The standard digraph for an n -links tree-topology multibody system is a tree with $(n+1)$ nodes, as illustrated in Figure 8. We

make no specific assumptions about the indexing scheme for the bodies. However, as usual, structural properties are easier to visualize when canonical indexing schemes are used.

Examining the kinematic velocity relationships across the links, the inter-link velocity relationship shown in Eq. 48 continues to apply to bodies in a tree-topology system:

$$\mathcal{V}(k) = \phi^*(\varphi(k), k) \mathcal{V}(\varphi(k)) + H^*(k) \dot{\theta}(k) \quad (49)$$

That is, the spatial velocity of the k^{th} link can be expressed as the sum of the rigidly propagated spatial velocity of the parent body and the relative spatial velocity, $H^*(k) \dot{\theta}(k)$, across the k^{th} hinge. Defining the \mathcal{V} and $\dot{\theta}$ stacked vectors and the H spatial operator in the same way as in Section 6.1.1, it is easy to verify that the following operator expression is a system-level rearrangement of the component-level velocity relationship in Eq. 49:

$$\mathcal{V} = \mathcal{E}_\phi^* \mathcal{V} + H^* \dot{\theta} \quad (50)$$

with \mathcal{E}_ϕ defined as

$$\mathcal{E}_\phi = \sum_{k=1}^n e_{\varphi(k)} \phi(\varphi(k), k) e_k^* \quad (51)$$

The \mathcal{E}_ϕ spatial operator has the Eq. 13 form of a BWA matrix for the SKO-forest associated with the system. The $\phi(\mathcal{P}(k), k)$ matrices are the 6×6 weight matrices. Since the SKO-forest is a forest, \mathcal{E}_ϕ is nilpotent and, from Lemma 4, it has a well defined 1-resolvent, $\phi = (\mathbf{I} - \mathcal{E}_\phi)^{-1}$. Hence, Eq. 50 can be transformed into

$$(\mathbf{I} - \mathcal{E}_\phi^*) \mathcal{V} = H^* \dot{\theta} \quad \text{and} \quad \mathcal{V} = \phi^* H^* \dot{\theta} \quad (52)$$

This operator expression is identical to the one in Eq. 34 for serial-chain systems.

7.1 Equations of motion for tree topology systems

To develop the equations of motion for the tree system, we once again start with the free-body equations of motion of the k^{th} link, as was done in Eq. 41 for serial-chain systems. The key difference is that we need to take into consideration the f_j spatial interaction forces from all the children bodies, instead of from just a single child body, as is the case for serial-chain systems. The following is the generalization of the force balance expression from Eq. 41 extended to tree-topology systems:

$$\ddot{\mathbf{r}}(k) - \sum_{\forall j \in \mathcal{C}(k)} \phi(k, j) \ddot{\mathbf{r}}(j) = M(k) \alpha(k) + \mathbf{b}(k) \quad (53)$$

Rearranging the above leads to the following analog of Eq. 42:

$$\ddot{\mathbf{r}}(k) - \sum_{\forall j \in \mathcal{C}(k)} \phi(k, j) \ddot{\mathbf{r}}(j) = M(k) \alpha(k) + \mathbf{b}(k) \quad (54)$$

Switching to the system-level stacked vector form of Eq. 54, we obtain

$$\dot{\mathbf{f}} = \mathcal{E}_\phi^* \dot{\mathbf{f}} + \mathbf{M}\alpha + \mathbf{b} \quad (55)$$

Continuing in this mode, the following expressions summarize the operator expressions for tree-topology systems:

$$\begin{aligned} \mathcal{V} &= \mathcal{E}_\phi^* \mathcal{V} + H^* \dot{\theta} \\ \alpha &= \mathcal{E}_\phi^* \alpha + H^* \ddot{\theta} + \mathbf{a} \\ \dot{\mathbf{f}} &= \mathcal{E}_\phi^* \dot{\mathbf{f}} + \mathbf{M}\alpha + \mathbf{b} \\ \mathcal{T} &= H\dot{\mathbf{f}} \end{aligned} \quad (56)$$

These expressions are identical to the corresponding expressions in Eq. 44 for serial-chain systems. Using $\phi = (\mathbf{I} - \mathcal{E}_\phi)^{-1}$, these expressions can be transformed from implicit ones into the following explicit operator expressions:

$$\begin{aligned} \mathcal{V} &= \phi^* H^* \dot{\theta} \\ \alpha &= \phi^* (H^* \ddot{\theta} + \mathbf{a}) \\ \dot{\mathbf{f}} &= \phi^* (\mathbf{M}\alpha + \mathbf{b}) \\ \mathcal{T} &= H\dot{\mathbf{f}} \end{aligned} \quad (57)$$

These expressions are identical to the corresponding expressions in Eq. 44 for serial-chain systems. Combining the expressions in Eq. 57 leads to

$$\mathcal{T} = \mathcal{M}(\theta) \ddot{\theta} + \mathcal{C}(\theta, \dot{\theta}) \quad (58)$$

where

$$\mathcal{M}(\theta) = H\phi\mathbf{M}\phi^*H^* \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}} \quad \text{and} \quad \mathcal{C}(\theta, \dot{\theta}) \triangleq H\phi(\mathbf{M}\phi^*\mathbf{a} + \mathbf{b}) \in \mathbb{R}^{\mathcal{N}} \quad (59)$$

$\mathcal{M} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}}$ denotes the mass matrix for the tree-topology system, and $\mathcal{C} \in \mathbb{R}^{\mathcal{N}}$ is the vector of velocity dependent nonlinear Coriolis and gyroscopic velocity dependent terms. These operator expressions are identical to the ones in Eq. 47 for serial-chain systems. Adopting the terminology from the serial-chain case, we refer to the expression for \mathcal{M} in Eq. 59 as the Newton-Euler Factorization of the mass matrix.

We thus conclude that, even though the component-level relationships and the respective operator structures are quite different between canonical and non-canonical serial-chain systems and tree-topology systems, the BWA and 1-resolvent matrix properties of the \mathcal{E}_ϕ and ϕ spatial operators persist, and the operator level form of the equations of motion and the mass matrix, remarkably, remain unchanged!

7.1.1 SKO and SPO operators—Now for some change in terminology. BWA matrices and 1-resolvent matrices will start to play a central role in our ongoing study of the dynamics of multibody systems. In the narrower context of multibody systems, we have already encountered spatial operator examples that are in fact BWA and 1-resolvent matrices. In order to conform more closely to the spatial operator terminology, we will henceforth in this multibody context refer to spatial operators that are BWA matrices (such

as \mathcal{E}_ϕ) as **spatial kernel operators (SKO)** and their associated 1-resolvent matrices (such as ϕ) as **spatial propagation operators (SPO)**.

8 Illustration of extensions for flexible bodies

The multibody role of SKO and SPO operators has so far focused on rigid body multibody systems. Spatial operator techniques, and analytical results such as Eq. 1, have been developed for systems with deformable bodies [18]. For the purposes of illustration, we now present an overview of the extensions that establish the SKO and SPO properties of the spatial operators for systems with non-rigid bodies using the notation from reference [18].

1. In addition to the 6-dimensional $\mathcal{V}(k)$ spatial velocity of its body frame, the velocity state of a body is defined by its $n_{md}(k)$ dimensional $\dot{\mathbf{q}}(k)$ deformation rate coordinates. Thus, the $\tilde{\mathcal{V}}(k) = n_{md}(k) + 6$ dimensional free-body generalized velocities vector $\mathcal{V}_{fl}(k)$ for the k^{th} body is defined as:

$$\mathcal{V}_{fl}(k) = \begin{bmatrix} \dot{\mathbf{q}}(k) \\ \mathcal{V}(k) \end{bmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(k)}$$

2. The appropriate inertia term for the k^{th} flexible body is defined by the following partitioned matrix

$$M_{fl}(k) = \begin{pmatrix} M_{fl}^{ff}(k) & M_{fl}^{fr}(k) \\ M_{fl}^{rf}(k) & M_{fl}^{rr}(k) \end{pmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(k) \times \tilde{\mathcal{N}}(k)}$$

where the sub-blocks define the rigid/rigid, rigid/flex and flex/flex inertia terms. The kinetic energy of the k^{th} body, including both the rigid and deformation contributions, is given by:

$$\mathcal{R}_e(k) = \frac{1}{2} \mathcal{V}_{fl}^*(k) M_{fl}(k) \mathcal{V}_{fl}(k)$$

3. The velocity recursion relating the $M_{fl}(k)$ velocity of the k^{th} body to that of its parent is expressed as

$$\mathcal{V}_{fl}(k) = \Phi_{fl}^*(k+1, k) \mathcal{V}_{fl}(k+1) + H_{fl}^*(k) \dot{\boldsymbol{\vartheta}}(k)$$

The inter-body transformation operator $\Phi_{fl}(k+1, k)$, the modal joint map matrix $H_{fl}(k)$, and the independent $\boldsymbol{\vartheta}(k)$ generalized coordinates, for the k^{th} body are defined as:

$$\Phi_{fl(k+1,k)} \triangleq \begin{pmatrix} \mathbf{0} & \Pi^*(\mathbb{O}_{k+1}^+) \phi(\mathbb{O}_k^+, k) \\ \mathbf{0} & \phi(k+1, k) \end{pmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(k+1) \times \tilde{\mathcal{N}}(k)} \quad H_{fl}(k) \triangleq \begin{pmatrix} \mathbf{I} & -\Pi_{\mathbb{B}}^*(\mathbb{O}_k) \\ \mathbf{0} & H_{\mathbb{B}}(k) \end{pmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(k) \times \tilde{\mathcal{N}}(k), \boldsymbol{\vartheta}(k)} \triangleq \begin{bmatrix} \eta(k) \\ \theta(k) \end{bmatrix} \in \mathcal{R}^{\tilde{\mathcal{N}}(k)}$$

The $n_{md}(k) \times 6$ dimensional $\Pi^*(\mathbb{O}_{k+1}^+)$ and $\Pi_{\mathbb{B}}^*(\mathbb{O}_k)$ modal influence matrices are obtained from the structural analysis of the body and define the mapping to the deformation of the nodes on either side of the k^{th} hinge. From this, the SKO weight matrices are identified to be the $\Phi_{fl}(k+1, k)$ terms.

4. The \mathcal{V}_{fl} and \mathcal{D} stacked vectors are assembled as usual. The $\mathcal{E}_{\Phi_{fl}}$ SKO operator is defined with the $\Phi_{fl}(k+1, k)$ weight matrices. The SPO operator is defined by $\Phi_{fl} = (\mathbf{I} - \mathcal{E}_{\Phi_{fl}})^{-1}$. The block-diagonal H_{fl} and M_{fl} operators are defined using the $H_{fl}(k)$ and $M_{fl}(k)$ component matrices. These lead to the following operatorlevel equations of motion

$$\begin{aligned}\mathcal{V}_{fl} &= \Phi_{fl}^* H_{fl}^* \mathcal{D} \\ \alpha_{fl} &= \Phi_{fl}^* (H_{fl}^* \mathcal{D} + \alpha_{fl}) \\ \mathfrak{f}_{fl} &= \Phi_{fl} (M_{fl} \alpha_{fl} + \mathfrak{b}_{fl} + \mathcal{R} \mathcal{D}) \\ \mathcal{T}_{fl} &= H_{fl} \mathfrak{f}_{fl} = \mathcal{M}_{fl} \ddot{\mathcal{D}} + \mathcal{C}_{fl}\end{aligned}$$

with the SKO formulation mass matrix and its Newton-Euler Factorization defined as

$$\mathcal{M}_{fl} \triangleq H_{fl} \Phi_{fl} M_{fl} \Phi_{fl}^* H_{fl}^* \quad \text{and} \quad \mathcal{C}_{fl} \triangleq H_{fl} \Phi_{fl} (M_{fl} \Phi_{fl}^* \alpha_{fl} + \mathfrak{b}_{fl} + \mathcal{R} \mathcal{D})$$

9 Conclusions

This paper develops insights into the graph theory underpinnings of multibody dynamics. We introduce the notion of BWA matrices as weighted matrix generalizations of digraph adjacency matrices. The BWA matrices for trees are shown to be nilpotent, thus making possible explicit expressions for their 1-resolvent. Key connections between BWA matrices, their 1-resolvents and spatial operators used for formulating the dynamics of tree multibody systems are subsequently established. In the dynamics context, the BWA matrices and their 1-resolvents are referred to as SKO and SPO operators respectively. The SKO operators provide a uniform and general way of formulating the system dynamics, with the system-specific details being reflected in the component weight matrices of the operator. The separation between the abstract SKO structure and the specific weight matrices is important because we show in the companion paper Jain [14] that key analytical results are a direct consequence of the SKO properties of the system, and are in fact independent of the specific nature and details of the component weight matrices for the system.

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.⁵ This project was also supported in part by Grant Number RO1GM082896-01A2 from the National Institute of Health.

References

1. Aghili, F. Simplified Lagrangian Mechanical Systems with Constraints using Square-root Factorization. Multibody Dynamics; ECCOMAS Thematic Conference; 2007. 2007
2. Aghili, F. A Gauge-invariant Formulation for Constrained Robotic Systems using Square-root Factorization and Unitary Transformation; IEEE/RSJ Int. Conf. on Intelligent Robots and Systems; 2008. p. 2814-2821.
3. Arczewski K. Determination of Christoffels symbols of the first kind for a class of rigid body systems by topological considerations. Int J Mech Sci. 1986; 28(1):11–22.
4. Arczewski K. Application of Graph Theory to the Mathematical Modelling of a Class of Rigid Body Systems. J of the Franklin Inst. 1990; 327(2):209–223.

⁵© 2010 California Institute of Technology. Government sponsorship acknowledged.

5. Baraff, D. Proc. ACM SIGGRAPH '96. New Orleans: 1996. Linear-Time Dynamics using Lagrange Multipliers; p. 137-146.
6. Cherowitzo W. Graph and Digraph Glossary. 2009 Website, URL <http://wwwmath.cudenver.edu/wcherowi/courses/m4408/glossary.htm/>.
7. Featherstone, R. Rigid Body Dynamics Algorithms. Springer Verlag; 2008.
8. Featherstone, R.; Orin, D. IEEE International Conference on Robotics and Automation. San Francisco, CA: 2000. Robot Dynamics: Equations and Algorithms; p. 826-834.
9. Gross, J.L.; Yellen, J. Graph theory and its applications. CRC Press; 2006.
10. Herman P, Kozlowski K. A Survey of Equations of Motion in terms of Inertial Quasi-velocities for Serial Manipulators. Arch Appl Mech. 2006; 76:579–614.
11. Jain A. Unified Formulation of Dynamics for Serial Rigid Multibody Systems. Journal of Guidance, Control and Dynamics. 1991; 14(3):531–542.
12. Jain, A. Correction to 'Spatial Operator Algebra for Multibody System Dynamics'. 2009. <http://dartslab.jpl.nasa.gov/References/pdf/gen-mbody-correction.pdf>
13. Jain, A. Recursive Algorithms Using Local Constraint Embedding for Multibody System Dynamics. ASME; International Design Engineering Technical Conferences & Computers and Information in Engineering Conference; San Diego, CA. 2009. 2009
14. Jain A. Graph Theoretic Foundations of Multibody Dynamics: Part II - Analysis and Algorithms. 2010 In review.
15. Jain, A. Graph Theory Roots of Spatial Operators for Kinematics and Dynamics; IEEE International Conference on Robotics and Automation; Anchorage, Alaska. 2010. 2010
16. Jain, A. Robot and Multibody Dynamics: Analysis and Algorithms. New York: Springer; 2010.
17. Jain, A.; Rodriguez, G. Recursive Dynamics for Geared Robotic Manipulators; IEEE Conference on Decision and Control; Honolulu, Hawaii. 1990.
18. Jain A, Rodriguez G. Recursive Flexible Multibody System Dynamics Using Spatial Operators. Journal of Guidance, Control and Dynamics. 1992; 15(6):1453–1466.
19. Jain A, Rodriguez G. An Analysis of the Kinematics and Dynamics of Under-actuated Manipulators. IEEE Transactions on Robotics and Automation. 1993; 9(4):411–422.
20. Jain A, Rodriguez G. Diagonalized Lagrangian Robot Dynamics. IEEE Transactions on Robotics and Automation. 1995; 11(4):571–584.
21. Jain A, Rodriguez G. Multibody Mass Matrix Sensitivity Analysis Using Spatial Operators. Journal for Multiscale Computational Engineering. 2003; 1(2–3)
22. Junkins JL, Schaub H. An Instantaneous Eigenstructure Quasivelocity Formulation for Nonlinear Multibody Dynamics. The Journal of Astronautical Sciences. 1997; 45(3):279–295.
23. Kreutz-Delgado K, Jain A, Rodriguez G. Recursive Formulation of Operational Space Control. The International Journal of Robotics Research. 1992; 11(4):320–328.
24. McPhee JJ, Ishac MG, Andrews GC. Wittenburg's formulation of multibody dynamics equations from a graph-theoretic perspective. Mechanism and machine theory. 1996; 31(2):201–213.
25. 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.
26. Ramoni, MF. Bayesian Networks Representation and Reasoning. 2009. URL <http://ocw.mit.edu/NR/rdonlyres/Health-Sciences-and-Technology/HST-951JMedical-Decision-SupportSpring2003/695229E8-B5C8-4F33-A5B9-C4079AFAF967/0/lecture5.pdf>
27. Rodriguez G. Kalman Filtering, Smoothing and Recursive Robot Arm Forward and Inverse Dynamics. IEEE Journal of Robotics and Automation. 1987; 3(6):624–639.
28. 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.
29. Rodriguez G, Jain A, Kreutz-Delgado K. Spatial Operator Algebra for Multibody System Dynamics. Journal of the Astronautical Sciences. 1992; 40(1):27–50.
30. Saha SK. A Decomposition of the Manipulator Inertia Matrix. IEEE Transactions on Robotics and Automation. 1997; 13(2):301–304.

31. Shi P, McPhee JJ. Dynamics of Flexible Multibody Systems using Virtual Work and Linear Graph Theory. *Multibody System Dynamics*. 2000; 4:355–381.
32. Siciliano, B.; Khatib, O. *Springer Handbook of Robotics*. Springer; 2008.
33. Sinclair AJ, Hurtado JE, Junkins JL. Linear feedback control using quasi velocities. *Journal of Guidance, Control, and Dynamics*. 2006; 29(6):1309–1314.
34. West, DB. *Introduction to Graph Theory*. Prentice Hall; 2001.
35. Wikipedia. Tree (graph theory). 2009. URL [http://en.wikipedia.org/w/index.php?title=Tree_\(graph_theory\)&oldid=327865637](http://en.wikipedia.org/w/index.php?title=Tree_(graph_theory)&oldid=327865637)
36. Wittenburg, J. *Dynamics of Systems of Rigid Bodies*. Stuttgart: B. G. Teubner; 1977.
37. Zanganeh K, Angeles J. A Formalism for the Analysis and Design of Modular Kinematic Structures. *International Journal of Robotics Research*. 1998; 17(7):720–730.

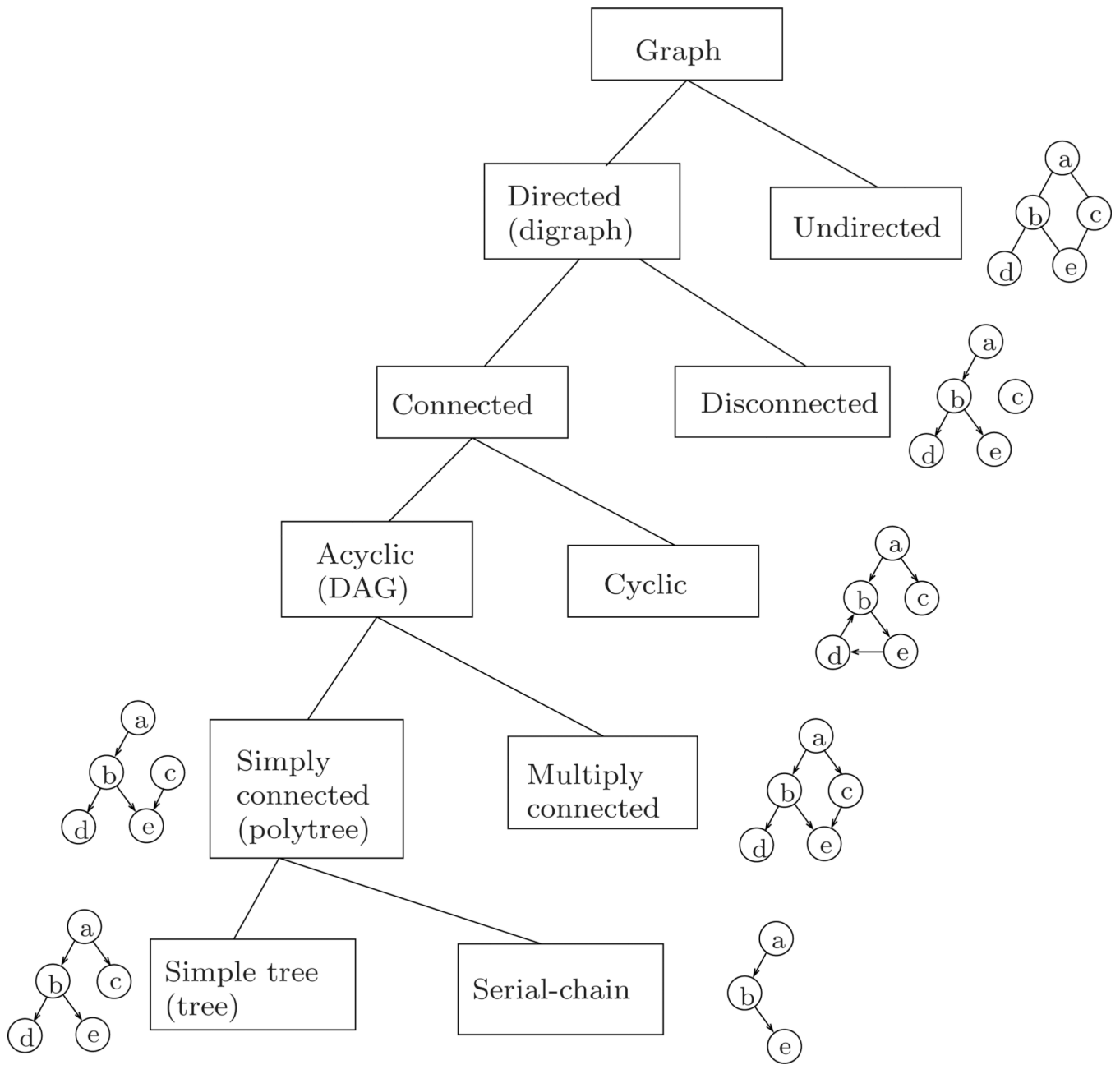


Fig. 1.
Hierarchical classification of graphs and digraphs

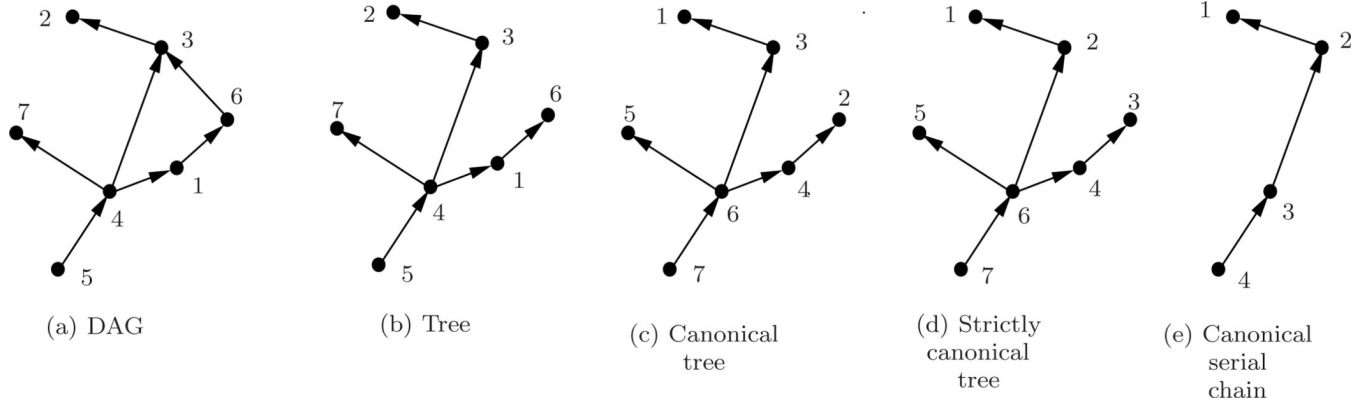


Fig. 2. Illustrations a digraphs with a DAG, a tree, canonical and strictly canonical trees and a serial-chain.

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{1} & \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \mathbf{1} \\ \cdot & \cdot & \cdot & \mathbf{1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

(a) DAG

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{1} & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{1} & \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \mathbf{1} \\ \cdot & \cdot & \cdot & \mathbf{1} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

(b) Tree

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{1} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \mathbf{1} & \mathbf{1} & \mathbf{1} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{1} & \cdot \end{pmatrix}$$

(c) Canonical tree

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{1} & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \mathbf{1} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \mathbf{1} & \cdot & \mathbf{1} & \mathbf{1} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{1} \end{pmatrix}$$

(d) Strictly canonical tree

$$\begin{pmatrix} \cdot & \cdot & \cdot & \cdot \\ \mathbf{1} & \cdot & \cdot & \cdot \\ \cdot & \mathbf{1} & \cdot & \cdot \\ \cdot & \cdot & \mathbf{1} & \cdot \end{pmatrix}$$

(e) Canonical serial-chain

Fig. 3.
The adjacency matrices for the digraphs illustrated in Figure 2.

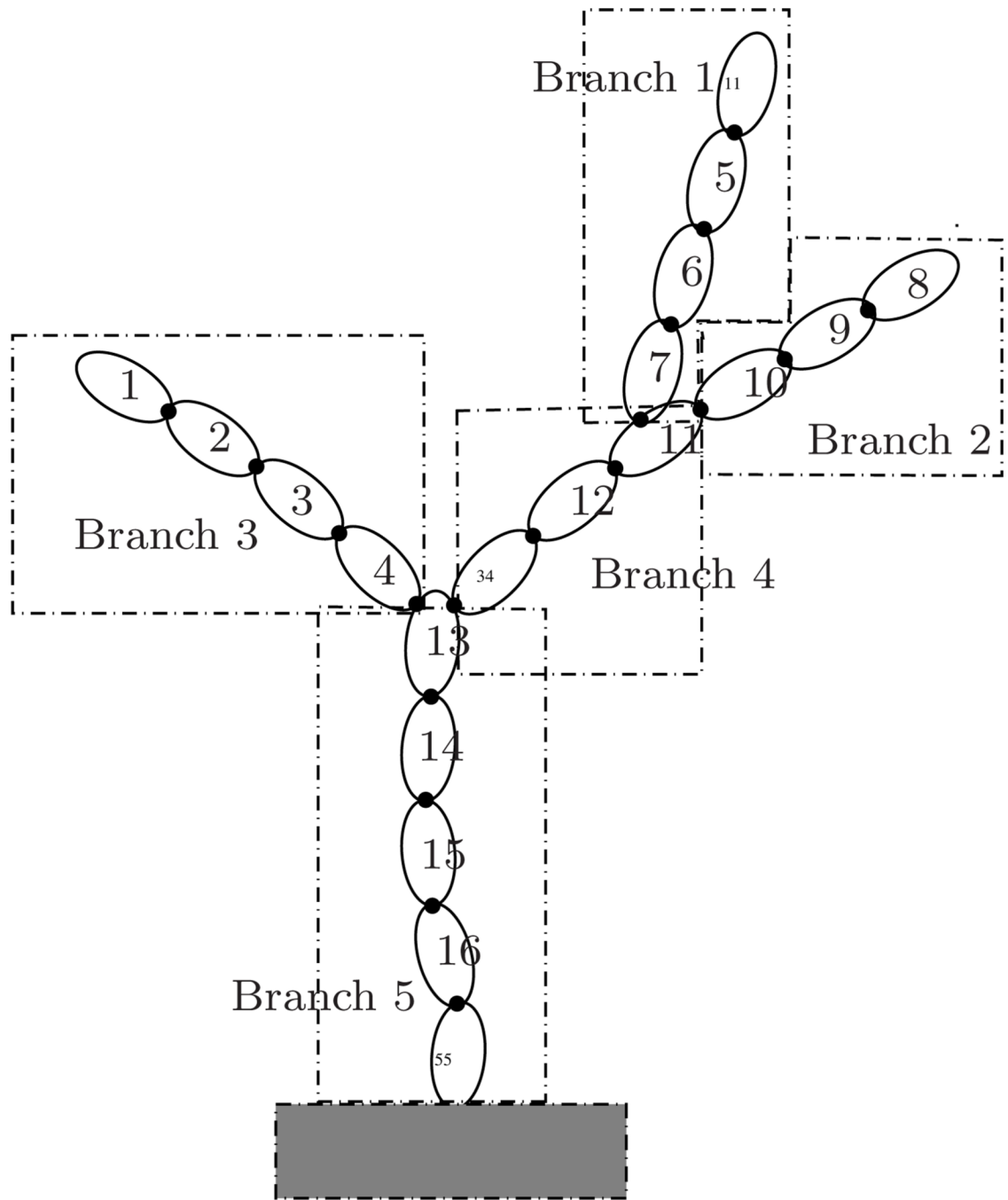


Fig. 4. An example strictly-canonical tree topology system decomposed into serial-chain branch segments.

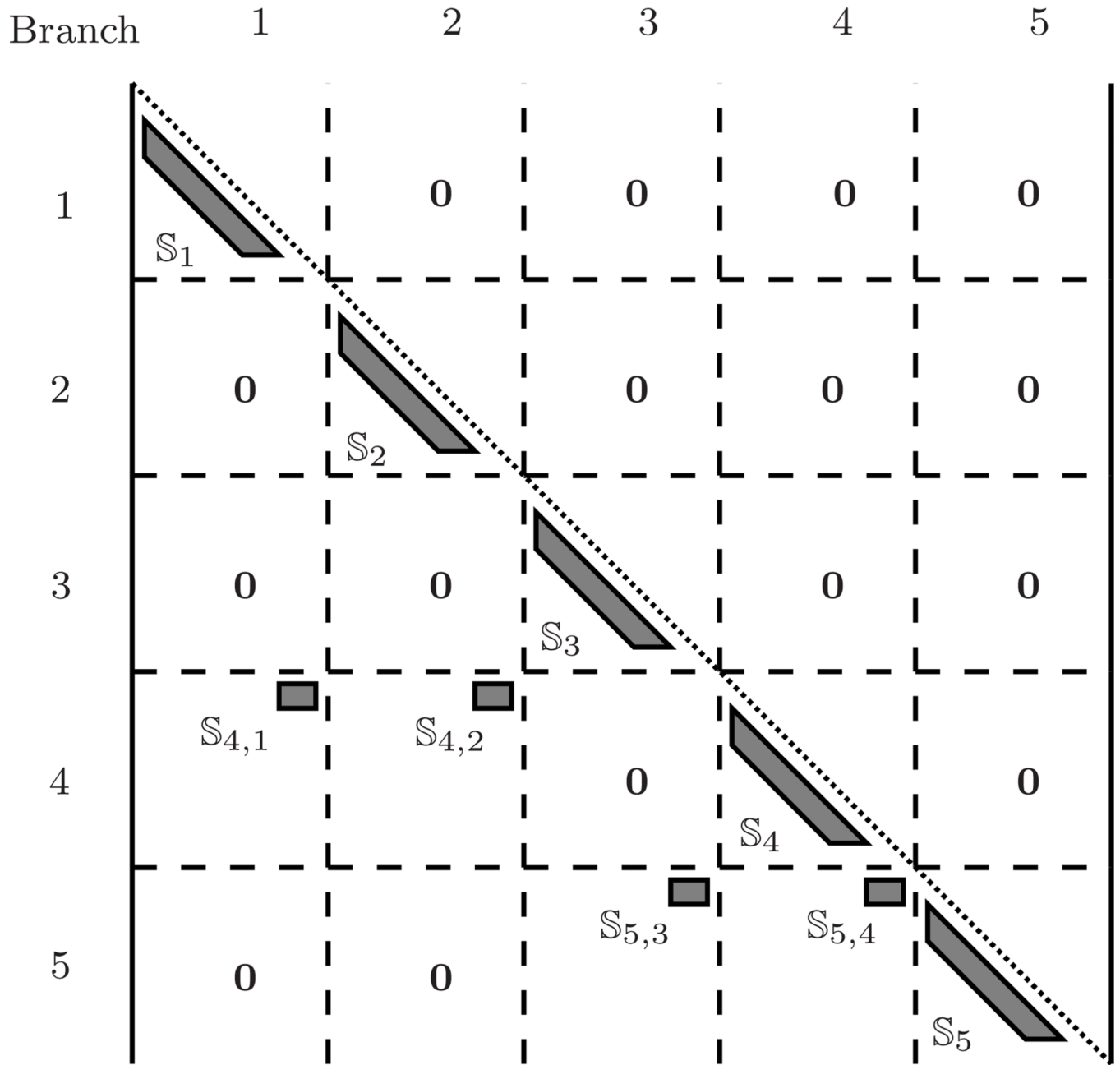


Fig. 5. Structure of the Sadjacency matrix for the strictly-canonical tree topology system in Figure 4

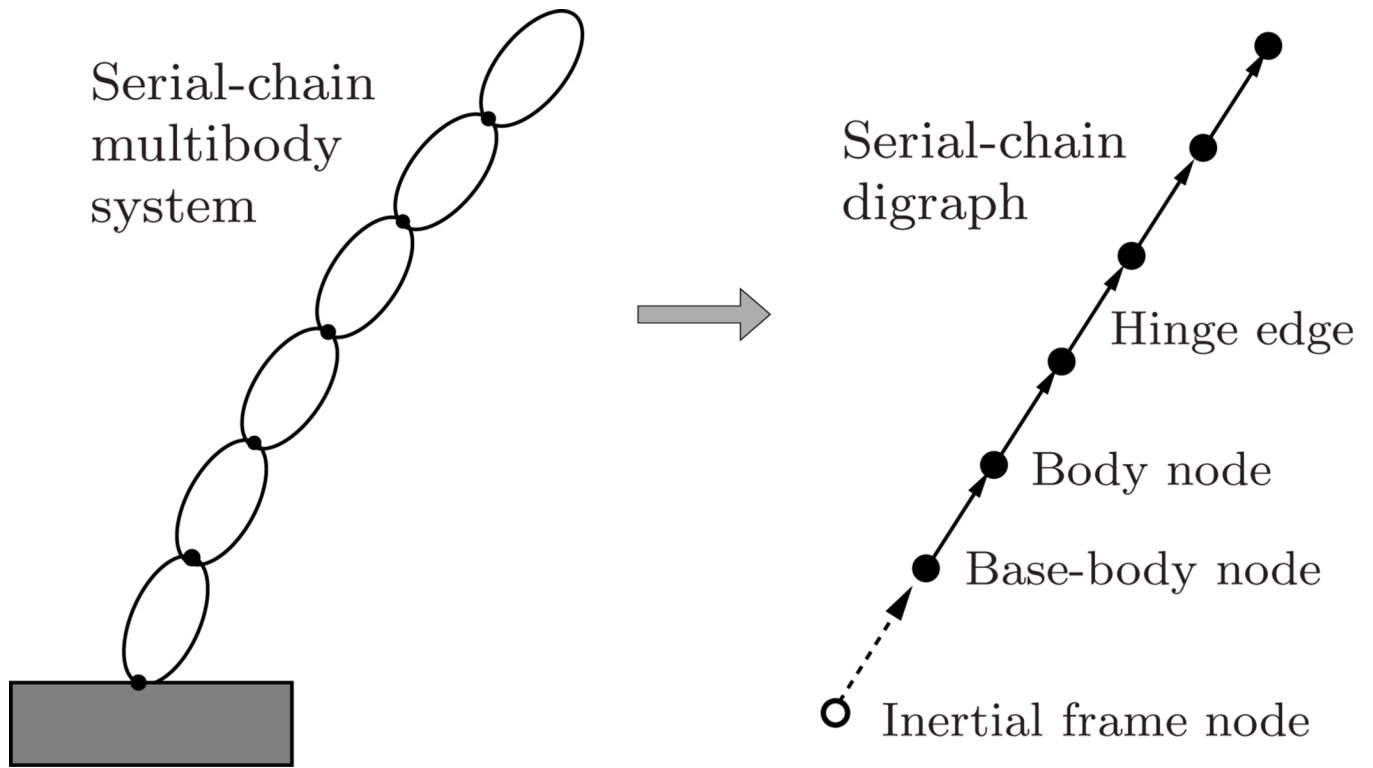
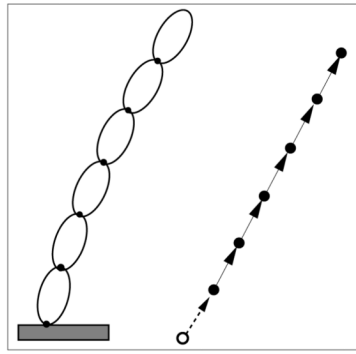
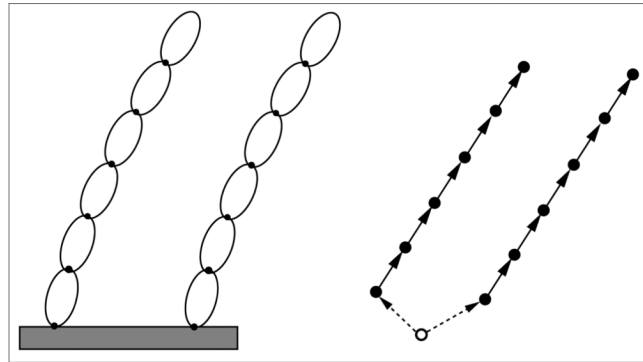


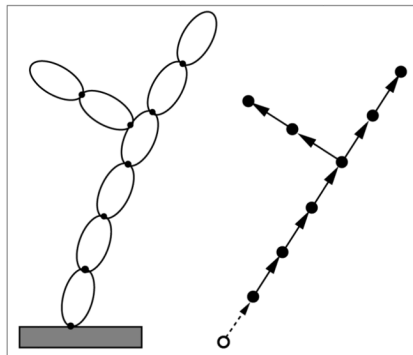
Fig. 6. The relationship between a serial-chain multibody system and its standard, serial-chain digraph. The inertial frame nodes are shown as unfilled nodes, while the edges from the node are shown as dashed lines.



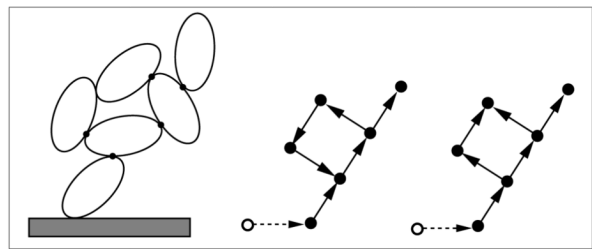
(1) A serial manipulator (serial-chain digraph)



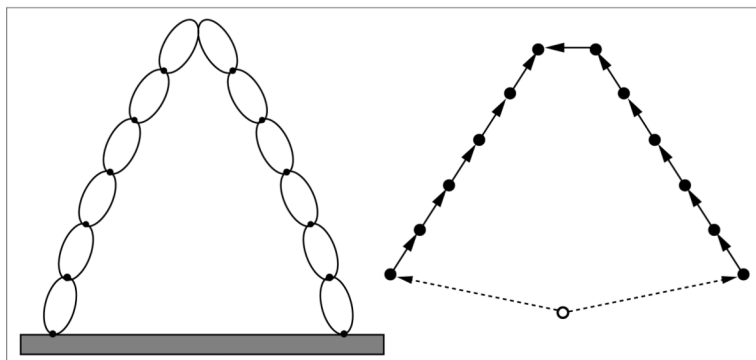
(2) Pair of independent serial manipulators (tree digraph)



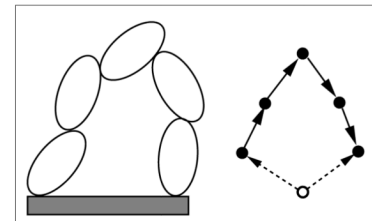
(3) Torso with arms (tree digraph)



(4) Mechanism with a four-bar linkage (digraph with directed cycle and a multiply-connected digraph)



(5) Pair of constrained serial manipulators (multiply-connected digraph)



(6) Serial manipulator with constrained end-effector (multiply-connected digraph)

Fig. 7. Examples of general multibody systems and their standard digraph representations.

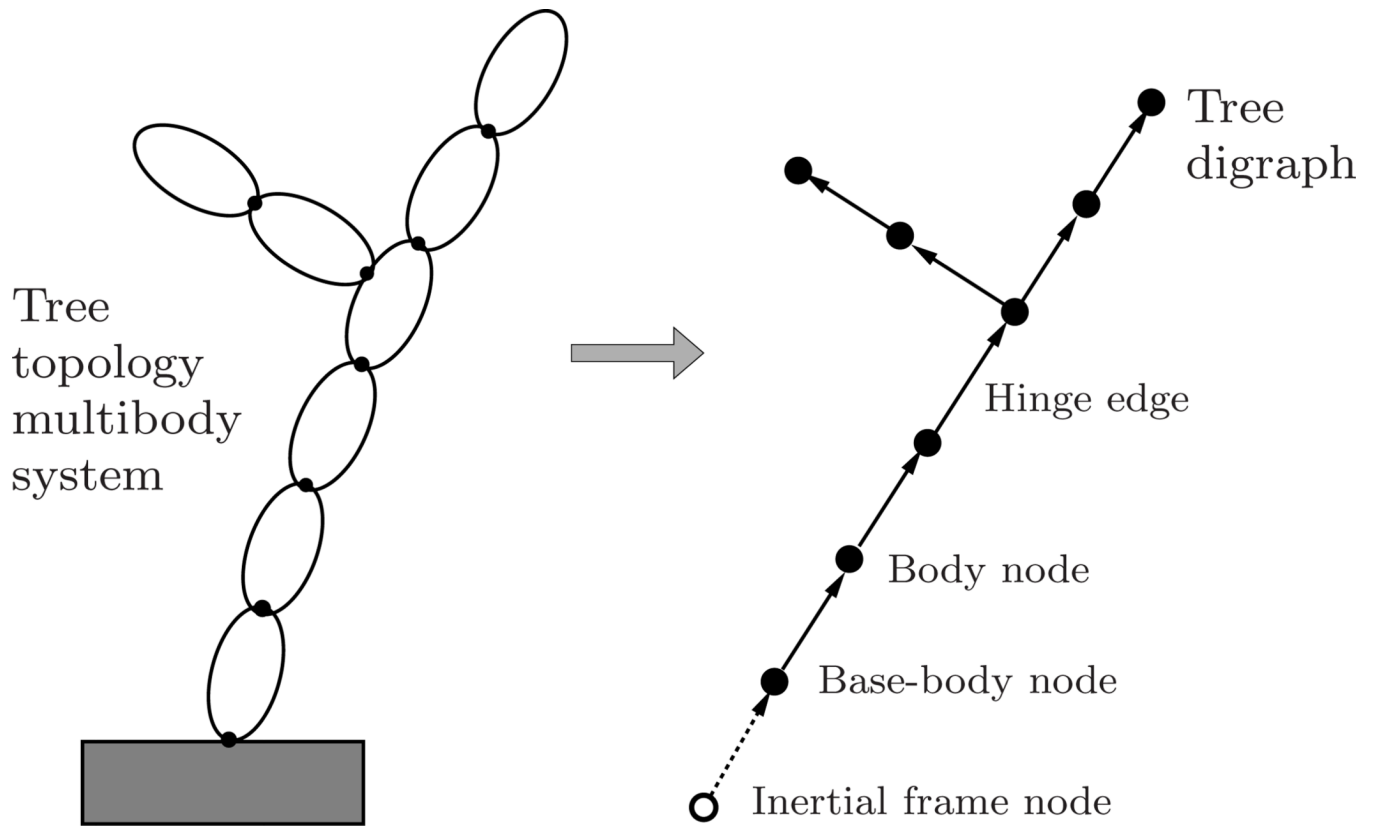


Fig. 8. The correspondence between a tree-topology multibody system and its standard tree digraph.