The 5th Joint International Conference on Multibody System Dynamics
June 24 – 28, 2018, Lisboa, Portugal

# Computing Inter-Body Constraint Forces in Recursive Multibody Dynamics

Abhinandan Jain[1]
[1]*Mobility and Robotic Systems, Jet Propulsion Laboratory/California Institute of Technology, jain@jpl.nasa.gov*

ABSTRACT *Minimal coordinate dynamics models are able to use recursive methods that avoid the need for Lagrange multipliers while solving the equations of motion. This approach is made possible by the rich underlying structure of the dynamics model that allows for the analytical factorization and inversion of the mass matrix. The absence of the Lagrange multipliers also means that inter-body constraint forces are not computed as part of the solution process. The general misconception is that additional expensive computations are needed to compute these constraint forces, and that these additional costs overcome the computational advantages of the recursive methods. In this paper we address this criticism of the recursive methods and show that they are completely unfounded. We show that there are simple and very low cost methods available to compute the constraint forces that directly use articulated body algorithm quantities that are by products of the recursive solution process.*

## 1 Introduction

The choice of a dynamics modeling approach is a key step towards solving the equations of motion of multibody systems for time-domain simulations. A popular choice of a dynamics model is the non-minimal coordinates approach that uses absolute coordinates for the individual bodies, and treats the inter-body hinges as explicit bilateral constraints for the system dynamics. This approach uses Lagrange multipliers to compute the inter-body constraint forces as part of the solution process, but requires a DAE solver to integrate the non-minimal coordinates.

An alternative approach is to use minimal coordinate dynamics models. The recursive methods for this approach avoid constraints and the need for Lagrange multipliers, and directly solve for the generalized accelerations. This is made possible by exploiting the rich underlying structure that allows for the analytical factorization and inversion of the mass matrix [1]. Despite the availability of these faster O(N) recursive algorithms [1, 2] for solving the equations of motion, and the ability to use simpler ODE solvers, the added complexity of such models has been a deterrence to their wider use. We focus in this paper on a perceived shortcoming for these methods in the omission of the computation of inter-body constraint forces in the solution process when at times they are needed for monitoring internal stresses or for computing frictional forces. It is generally believed that additional expensive computations are needed to compute these constraint forces, and that these additional costs overcome the computational advantages of the recursive methods.

In this paper we address this criticism of the recursive methods and show that they are completely unfounded. We show that there are simple and very low cost methods available to compute the constraint forces should the need arise. The methods directly use the articulated body algorithm quantities that are by products of the recursive solution process. The main expression for computing the inter-body constraint forces has the form

$$\mathfrak{f}(\mathrm{k}) = \mathcal{P}(\mathrm{k})\mathfrak{z}(\mathrm{k}) + \alpha(\mathrm{k}) \tag{1}$$

Here $\mathfrak{f}(\mathrm{k})$ denotes the inter-body constraint spatial force between the $\mathrm{k}^{\mathrm{th}}$ body and its parent, while $\mathcal{P}(\mathrm{k})$, $\mathfrak{z}(\mathrm{k})$ and $\alpha(\mathrm{k})$ are articulated body quantities available from the recursions used in solving the equations of motion. This expression provides a very inexpensive way to compute the constraint force - and needs only be used only when

such forces are explicitly needed! While Eq. 1 is not new for tree-topology rigid body multibody systems [1, 3], it is not well known, leading to the above mentioned misconceptions about the recursive methods.

In this paper we look in further detail at the topic of computing constraint forces for multibody systems. For rigid, tree multibody systems we explain the basis for Eq. 1 and its derivation. We also derive additional useful variants of this expression.

We next examine the same topic of constraint forces for more general multibody systems. We begin by looking at the case when there are non-rigid flexible bodies in a tree-topology multibody system. We show that a form of Eq. 1 continues to apply in this case.

We next look at closed-chain systems. We pursue two paths for such systems. The first is the tree-augmented dynamics model, where the system is decomposed into a spanning tree together with additional cut-joint constraints. We derive an extension of the above approach for computing the inter-body constraint forces for such dynamics models.

An alternative dynamics modeling approach for closed-chain systems is the more recently developed constraint embedding approach. In this approach, the original graph topology is converted into a tree-topology system using body aggregation and compound bodies. The resulting minimal-coordinates dynamics model can be solved using a form of the standard recursive methods. We extend the above approach for solving for constraint forces to such constraint embedding based dynamics models.

## 2 Rigid-Body. Minimal Coordinate Tree Dynamics

The aim of this section is to briefly summarize the essential ideas underlying spatial operators leading up to the Newton-Euler Operator Factorization $\mathcal{M}(\theta) = H\phi M\phi^* H^*$ of the manipulator mass matrix. While this is done here for a serial graph manipulator, the factorization results apply to more general class of complex joint-connected mechanical systems, including tree configurations with flexible links and joints [4].

Consider a serial manipulator with $\mathcal{N}$ rigid links. The links are numbered in increasing order from tip to base. The outer-most link is link 1, and the inner-most link is link $\mathcal{N}$. The overall number of degrees-of-freedom for the manipulator is $\mathcal{N}$. There are two joints attached to the $k^{th}$ link. A coordinate frame $\mathbb{O}_k$ is attached to the inboard joint, and another frame $\mathbb{O}^+_{k-1}$ is attached to the outboard joint. Frame $\mathbb{O}_k$ is also the body frame for the $k^{th}$ link. The $k^{th}$ joint connects the $(k+1)^{st}$ and $k^{th}$ links, and its motion is defined as the motion of frame $\mathbb{O}_k$ with respect to frame $\mathbb{O}^+_k$. When applicable, the free-space motion of a manipulator is modeled by attaching a 6 degree-of-freedom joint between the base link and the inertial frame about which the free-space motion occurs. However, in this paper, without loss of generality and for the sake of notational simplicity, all joints are assumed to be single rotational degree-of-freedom joints with the $k^{th}$ joint coordinate given by $\theta(k)$. Extension to joints with more rotational and translational degrees-of-freedom is straightforward [5].

The transformation operator $\phi(k,k-1)$ between the $\mathbb{O}_{k-1}$ and $\mathbb{O}_k$ frames is

$$\phi(k,k-1) \;=\; \begin{pmatrix} I_3 & \tilde{l}(k,k-1) \\ 0 & I_3 \end{pmatrix} \in \mathcal{R}^{6\times 6}$$

where $l(k,k-1)$ is the vector from frame $\mathbb{O}_k$ to frame $\mathbb{O}_{(k-1)}$, and $\tilde{l}(k,k-1) \in \mathcal{R}^{3\times 3}$ is the skew–symmetric matrix associated with the cross-product operation. We use coordinate free notation for notational simplicity.

The spatial velocity of the $k^{th}$ body frame $\mathbb{O}_k$ is $V(k) = [\omega^*(k), v^*(k)]^* \in \mathcal{R}^6$, where $\omega(k)$ and $v(k)$ are the angular and linear velocities of $\mathbb{O}_k$. With $h(k) \in \mathcal{R}^3$ denoting the $k^{th}$ joint axis vector, $H(k) = [h^*(k), 0] \in \mathcal{R}^1 \times \mathcal{R}^6$ denotes the joint map matrix for the joint, and the relative spatial velocity across the $k^{th}$ joint is $H^*(k)\dot{\theta}(k)$. The spatial force of interaction $f(k)$ across the $k^{th}$ joint is $f(k) = [N^*(k), F^*(k)]^* \in \mathcal{R}^6$, where $N(k)$ and $F(k)$ are the moment and force components respectively. The $6\times 6$ spatial inertia matrix $M(k)$ of the $k^{th}$ link in the coordinate frame $\mathbb{O}_k$ is

$$M(k) \;=\; \begin{pmatrix} \mathscr{J}(k) & m(k)\tilde{p}(k) \\ -m(k)\tilde{p}(k) & m(k)I_3 \end{pmatrix}$$

where $m(k)$ is the mass, $p(k)\in\mathcal{R}^3$ is the vector from $\mathbb{O}_k$ to the $k^{th}$ link center of mass, and $\mathscr{J}(k)\in\mathcal{R}^{3\times3}$ is the rotational inertia of the $k^{th}$ link about $\mathbb{O}_k$. $I_3$ is the $3\times3$ identity matrix.

The recursive Newton–Euler equations of motion are [6, 7]

$$\left\{\begin{array}{rcl} & V(\mathcal{N}+1)=0; \qquad \alpha(\mathcal{N}+1)=0 & \\ \textbf{for } k &=& \mathcal{N}\cdots1 \\ V(k) &=& \phi^*(k+1,k)V(k+1)+H^*(k)\dot\theta(k) \\ \alpha(k) &=& \phi^*(k+1,k)\alpha(k+1)+H^*(k)\ddot\theta(k)+a(k) \\ \textbf{end loop} & & \end{array}\right. \tag{2}$$

$$\left\{\begin{array}{rcl} & f(0)=0 & \\ \textbf{for } k &=& 1\cdots\mathcal{N} \\ f(k) &=& \phi(k,k-1)f(k-1)+M(k)\alpha(k)+b(k) \\ \mathcal{T}(k) &=& H(k)f(k) \\ \textbf{end loop} & & \end{array}\right.$$

where $\mathcal{T}(k)$ is the applied moment at joint $k$. The nonlinear, velocity dependent terms $a(k)$ and $b(k)$ are respectively the Coriolis acceleration and the gyroscopic force terms for the $k^{th}$ link.

The "stacked" notation $\theta=\text{col}\left\{\theta(k)\right\}_{k=1}^{n}\in\mathcal{R}^{\mathcal{N}}$ is used to simplify the above recursive Newton-Euler equations. This notation [8] eliminates the arguments $k$ associated with the individual links by defining composite vectors, such as $\theta$, which apply to the entire system. We define

$$\begin{array}{ll} \mathcal{T}=\text{col}\left\{\mathcal{T}(k)\right\}_{k=1}^{n}\in\mathcal{R}^{\mathcal{N}} & V=\text{col}\left\{V(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}} \\ f=\text{col}\left\{f(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}} & \alpha=\text{col}\left\{\alpha(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}} \\ a=\text{col}\left\{a(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}} & b=\text{col}\left\{b(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}} \end{array}$$

In this notation, the equations of motion in Eq. 2 can be re-expressed as [7, 9]:

$$\begin{array}{ll} V=\phi^*H^*\dot\theta; & \alpha=\phi^*[H^*\ddot\theta+a] \\ f=\phi[M\alpha+b]; & \mathcal{T}=Hf=\mathcal{M}\ddot\theta+\mathcal{C} \end{array} \tag{3}$$

where the mass matrix $\mathcal{M}(\theta)=H\phi M\phi H^*$; $\mathcal{C}(\theta,\dot\theta)=H\phi[M\phi^*a+b]\in\mathcal{R}^{\mathcal{N}}$ is the Coriolis term; $H=\text{diag}\left\{H(k)\right\}_{k=1}^{n}\in\mathcal{R}^{\mathcal{N}\times6\mathcal{N}}$; $M=\text{diag}\left\{M(k)\right\}_{k=1}^{n}\in\mathcal{R}^{6\mathcal{N}\times6\mathcal{N}}$; and $\phi\in\mathcal{R}^{6\mathcal{N}\times6\mathcal{N}}$ is

$$\phi=(I-\mathcal{E}_\phi)^{-1}=\begin{pmatrix} I & 0 & \dots & 0 \\ \phi(2,1) & I & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \phi(n,1) & \phi(n,2) & \dots & I \end{pmatrix} \tag{4}$$

with $\phi(i,j)=\phi(i,i-1)\cdots\phi(j+1,j)$ for $i>j$. The shift operator $\mathcal{E}_\phi\in\mathcal{R}^{6\mathcal{N}\times6\mathcal{N}}$ is defined as

$$\mathcal{E}_\phi=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \phi(2,1) & 0 & \dots & 0 & 0 \\ 0 & \phi(3,2) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \phi(\mathcal{N},\mathcal{N}-1) & 0 \end{pmatrix} \tag{5}$$

Also the $\tilde{\phi}$ operator is defined as $\phi - \mathrm{I}$. It can be shown that [1]

$$\tilde{\phi} = \phi - \mathrm{I} = \mathcal{E}_\phi \phi = \phi \mathcal{E}_\phi \tag{6}$$

## 2.1 Mass Matrix Factorization and Inversion

The *articulated body inertia*, $\mathcal{P}(\mathrm{k})$, for the $\mathrm{k}^{\mathrm{th}}$ body can be obtained via the following tip-to-base recursive solution to a Riccati equation [1].

$$\left\{ \begin{array}{l} \qquad \mathcal{P}(0) = 0, \quad \overline{\tau}(0) = \mathbf{0} \\ \textbf{for } \mathrm{k} \;=\; 1 \cdots \mathcal{N} \\ \quad \psi(\mathrm{k}, \mathrm{k}-1) = \phi(\mathrm{k}, \mathrm{k}-1) \overline{\tau}(\mathrm{k}-1) \\ \qquad \mathcal{P}(\mathrm{k}) = \psi(\mathrm{k}, \mathrm{k}-1) \mathcal{P}(\mathrm{k}-1) \psi^*(\mathrm{k}, \mathrm{k}-1) + \mathrm{M}(\mathrm{k}) \\ \qquad \mathcal{D}(\mathrm{k}) = \mathrm{H}(\mathrm{k}) \mathcal{P}(\mathrm{k}) \mathrm{H}^*(\mathrm{k}) \\ \qquad \mathcal{G}(\mathrm{k}) = \mathcal{P}(\mathrm{k}) \mathrm{H}^*(\mathrm{k}) \mathcal{D}^{-1}(\mathrm{k}) \\ \quad \mathcal{K}(\mathrm{k}+1, \mathrm{k}) = \phi(\mathrm{k}+1, \mathrm{k}) \mathcal{G}(\mathrm{k}) \\ \qquad \overline{\tau}(\mathrm{k}) = \mathrm{I} - \mathcal{G}(\mathrm{k}) \mathrm{H}(\mathrm{k}) \\ \textbf{end loop} \end{array} \right. \tag{7}$$

The $\mathcal{P}()$ matrices play a key role in the recursive techniques for minimal coordinate dynamics. We can use it to define the block diagonal operator $\mathcal{P}$ as

$$\mathcal{P} \triangleq \mathrm{diag} \left\{ \mathcal{P}(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} \quad \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \tag{8}$$

and the following additional spatial operators:

$$\begin{array}{rcll} \mathcal{D} & \triangleq & \mathrm{diag} \left\{ \mathcal{D}(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} = \mathrm{H} \mathcal{P} \mathrm{H}^* & \in \mathcal{R}^{\mathcal{N} \times \mathcal{N}} \\ \mathcal{G} & \triangleq & \mathrm{diag} \left\{ \mathcal{G}(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} = \mathcal{P} \mathrm{H}^* \mathcal{D}^{-1} & \in \mathcal{R}^{6\mathrm{n} \times \mathcal{N}} \\ \mathcal{K} & \triangleq & \mathcal{E}_\phi \mathcal{G} & \in \mathcal{R}^{6\mathrm{n} \times \mathcal{N}} \\ \tau & \triangleq & \mathrm{diag} \left\{ \tau(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} = \mathcal{G} \mathrm{H} & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \\ \overline{\tau} & \triangleq & \mathrm{diag} \left\{ \overline{\tau}(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} = \mathrm{I} - \tau & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \\ \mathcal{P}^+ & \triangleq & \mathrm{diag} \left\{ \mathcal{P}^+(\mathrm{k}) \right\}_{\mathrm{k}=1}^{\mathrm{n}} = \overline{\tau} \mathcal{P} \overline{\tau}^* = \overline{\tau} \mathcal{P} = \mathcal{P} \overline{\tau}^* & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \\ \mathcal{E}_\psi & \triangleq & \mathcal{E}_\phi \overline{\tau} & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \\ \psi & \triangleq & (\mathrm{I} - \mathcal{E}_\psi)^{-1} & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \\ \tilde{\psi} & \triangleq & \psi - \mathrm{I} & \in \mathcal{R}^{6\mathrm{n} \times 6\mathrm{n}} \end{array} \tag{9}$$

The operators $\mathcal{D}$, $\mathcal{G}$, $\tau$, $\overline{\tau}$, and $\mathcal{P}^+$ are all block diagonal. The $\psi$ spatial operator shares the same structural properties as the $\phi$ operator.

Using these spatial operators one can obtain the following operator factorizations of the mass matrix and its inverse [1]:

$$\begin{array}{rcl} \mathcal{M} & = & \mathrm{H} \phi \mathrm{M} \phi^* \mathrm{H}^* \\ & = & [\mathrm{I} + \mathrm{H} \phi \mathcal{K}] \mathcal{D} [\mathrm{I} + \mathrm{H} \phi \mathcal{K}]^* \\ [\mathrm{I} + \mathrm{H} \phi \mathcal{K}]^{-1} & = & \mathrm{I} - \mathrm{H} \psi \mathcal{K} \\ \mathcal{M}^{-1} & = & [\mathrm{I} - \mathrm{H} \psi \mathcal{K}]^* \mathcal{D}^{-1} [\mathrm{I} - \mathrm{H} \psi \mathcal{K}] \end{array} \tag{10}$$

These mass matrix expressions have been derived and used extensively [3, 7–13], to develop a variety of spatially recursive algorithms for forward dynamics, for both rigid and flexible multibody systems of arbitrarily specified topologies, as well as closed-form analytical expressions for the inverse of the mass matrix. The spatial operators $\psi$, $\mathcal{D}$ correspond to a suitably defined spatially recursive Kalman filter, with the spatial operator $\mathcal{K}$ representing the Kalman gain for this filter. We also refer to these operators $\psi$, $\mathcal{D}$ and $\mathcal{K}$ as"articulated" quantities, because of their relationship to the articulated inertias first introduced by [14].

The mass matrix factor $[I + H\phi\mathcal{K}]$ is a square, invertible matrix and so is its inverse $[I - H\psi\mathcal{K}]$.

## 2.2 $O(\mathcal{N})$ Forward Dynamics

Using the expression for the mass matrix inverse in Eq. 10, and some additional spatial operator identities, it can be shown that [3]

$$\ddot{\theta} = \mathcal{M}^{-1}(\mathcal{T} - \mathcal{C}) = [I - H\psi\mathcal{K}]^* \mathcal{D}^{-1} [I - H\psi\mathcal{K}](\mathcal{T} - \mathcal{C}) \tag{11}$$

This expression can be broken down into the following sequence of simpler expressions:

$$\begin{array}{rcll} \mathfrak{z} & = & \psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b}) & \\ \epsilon & = & \mathcal{T} - H\mathfrak{z} & = \mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b}) \\ \nu & = & \mathcal{D}^{-1}\epsilon & = \mathcal{D}^{-1}[\mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})] \\ \mathfrak{z}^+ & = & \mathfrak{z} + \mathcal{G}^*\epsilon & \\ \alpha & = & \psi(H^*\nu + \mathfrak{a}) & = \psi^*(H^*\mathcal{D}^{-1}[\mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})] + \mathfrak{a}) \\ \alpha^+ & = & \mathcal{E}_\phi^*\alpha & \\ \ddot{\theta} & = & \nu - \mathcal{G}^*\alpha^+ & = [I - H\psi\mathcal{K}]^*\mathcal{D}^{-1}[\mathcal{T} - H\psi(\mathcal{K}\mathcal{T} + \mathcal{P}\mathfrak{a} + \mathfrak{b})] - \mathcal{K}^*\psi^*\mathfrak{a} \end{array} \tag{12}$$

These new operator expressions can be converted into recursive computational algorithms which do not require the explicit computation of the component operators. The resulting $O(\mathcal{N})$ forward dynamics procedure is as follows:

$$\left\{ \begin{array}{l} \qquad \mathfrak{z}(0) = 0, \quad \mathcal{T}(0) = 0 \\ \textbf{for } k = 1 \cdots \mathcal{N} \\ \quad \mathfrak{z}(k) = \psi(k, k-1)\mathfrak{z}(k-1) + \mathcal{P}(k)\mathfrak{a}(k) + \mathfrak{b}(k) + \mathcal{K}(k, k-1)T(k-1) \\ \quad \epsilon(k) = \mathcal{T}(k) - H(k)\mathfrak{z}(k) \\ \quad \nu(k) = \mathcal{D}^{-1}(k)\epsilon(k) \\ \textbf{end loop} \end{array} \right. \tag{13}$$

$$\left\{ \begin{array}{l} \qquad \alpha(n+1) = \mathbf{0} \\ \textbf{for } k = \mathcal{N} \cdots 1 \\ \quad \alpha^+(k) = \mathcal{E}_\phi^*\alpha(k+1) \\ \quad \ddot{\theta}(k) = \nu(k) - \mathcal{G}^*(k+1, k)\alpha^+(k) \\ \quad \alpha(k) = \overline{\tau}^*(k)\alpha^+(k+1) + H^*(k)\nu(k) + \mathfrak{a}(k) \\ \textbf{end loop} \end{array} \right.$$

This algorithm includes the recursive Riccati equation steps for the computation of the $\mathcal{P}(.)$ and $\mathfrak{z}(.)$ quantities. The fact that the computational cost of this algorithm is $O(\mathcal{N})$ follows from the fact that the computational cost of each of the steps in the above algorithm is fixed, and each of these steps is carried out $\mathcal{N}$ times during the course of the algorithm.

As we see here, this forward dynamics algorithm does not require the explicit computation of either $\mathcal{M}$ or $\mathcal{C}$. It does not require the explicit computation of any of the spatial operators either. It illustrates the ease with which the

high level operator level manipulations can be used to establish key identities and results, and at a later stage when the time for computations arises, these results can be mapped into highly efficient computational algorithms. While we have focused on serial-graph systems, all of the development in this section extend directly to tree-topology systems [5].

### 2.3 Computing Inter-Body Force

The AB forward dynamics algorithm described in the previous section does not explicitly compute the $\mathfrak{f}(k)$ body interaction spatial forces. These forces may be required in certain situations. One option is to use the body spatial acceleration computed in the forward dynamics algorithm to run the second tip-to-base recursion in the Newton-Euler inverse dynamics algorithm to compute these spatial forces. This is expensive and unnecessary. The following describes a better alternative based on the articulated body model that avoids the additional recursion and allows us to directly compute $\mathfrak{f}(k)$ for any link, as needed. The inter-link spatial forces $\mathfrak{f}$ are given by:

$$\mathfrak{f} = \mathcal{P}(\alpha - \mathfrak{a}) + \mathfrak{z} = \mathcal{P}^+\alpha^+ + \mathfrak{z}^+ \tag{14}$$

The derivation of the above expression involves several spatial operator transformations that are described in detail in [1] including the following:

$$\phi M \psi^* = \mathcal{P} + \tilde{\phi}\mathcal{P} + \mathcal{P}\tilde{\psi}^* \tag{15}$$

We paraphrase the derivation here and refer the reader to this reference for the full details.

$$\begin{aligned}
\mathfrak{f} &\overset{3}{=} \phi[M\alpha + \mathfrak{b}] \overset{12}{=} \phi\left[M\psi^*(H^*\nu + \mathfrak{a}) + \mathfrak{b}\right] \\
&\overset{15}{=} [\tilde{\phi}\mathcal{P} + \mathcal{P}\psi](H^*\nu + \mathfrak{a}) + \phi\mathfrak{b} \overset{12}{=} \tilde{\phi}\mathcal{P}H^*\nu + \tilde{\phi}\mathcal{P}\mathfrak{a} + \mathcal{P}\alpha + \phi\mathfrak{b} \\
&\overset{6,12}{=} \phi\mathcal{K}\epsilon + \phi\mathcal{P}\mathfrak{a} - \mathcal{P}\mathfrak{a} + \mathcal{P}\alpha + \phi\mathfrak{b} \overset{12}{=} \mathcal{P}[\alpha - \mathfrak{a}] + \mathfrak{z}
\end{aligned}$$

This establishes the first half of Eq. 14.
With $\alpha^+ = \mathcal{E}^*_\phi \alpha$, it can then be shown in [1] that $\alpha = \overline{\tau}^*\alpha^+ + H^*\nu + \mathfrak{a}$. Using the following identity [1]

$$\alpha = \overline{\tau}^*\alpha^+ + H^*\nu + \mathfrak{a} \tag{16}$$

it can be further shown that [1]

$$\mathfrak{f} \overset{14}{=} \mathcal{P}(\alpha - \mathfrak{a}) + \mathfrak{z} \overset{16}{=} \mathcal{P}[\overline{\tau}^*\alpha^+ + H^*\nu + \mathfrak{a}] - \mathcal{P}\mathfrak{a} + \mathfrak{z} \overset{9,12}{=} \mathcal{P}^+\alpha^+ + \mathcal{P}H^*\nu + \mathfrak{z}^+ - \mathcal{G}\epsilon \overset{12}{=} \mathcal{P}^+\alpha^+ + \mathfrak{z}^+ \tag{17}$$

The component expression for the $k^{th}$ body corresponding to Eq. 14 and Eq. 17 is:

$$\mathfrak{f}(k) = \mathcal{P}(k)(\alpha(k) - \mathfrak{a}(k)) + \mathfrak{z}(k) = \mathcal{P}^+(k)\alpha^+(k) + \mathfrak{z}^+(k) \tag{18}$$

It is noteworthy that these expressions for the inter-body force continue to hold even for prescribed hinges. The additional computational cost for evaluating the inter-body force is small once the equations of motion have been solved using the articulated body inertia algorithm. This is so because the required quantities in Eq. 18 are a by-product of this algorithm, and thus evaluating the inter-body force simply requires a matrix/vector 6-dimensional product and summing a couple of 6-vectors, and inverse dynamics computations are not needed. As we will see in the following sections, similar expressions continue to hold for more complex multibody systems involving flexible bodies and closed-chain topologies.

In the event that a hinge consists of multiple sub-hinges, such as universal or gimbal hinges, the inter-body spatial force value remains the same across them all (except for rigid body transformation for a prismatic sub-hinge) since there are no D'Alembert force "losses" from the mass-less sub-hinges.

# 3 Flexible Multibody Systems

We now turn to the case where the tree-topology system may have flexible bodies. Assumed modes provide a series expansion representation of the deformation field and are often used to represent the deformation of flexible bodies. A subset of the modes can be used for reduced order models for control system design. Lumped mass models are typically the starting point for developing assumed mode models.

There is a close relationship between the choice of a body reference frame and the type of assumed modes. The complete motion of the flexible body is defined by the motion of the body reference frame and the deformation of the body with respect to the body frame. In the multibody context, it is often convenient to choose the location of the $k^{th}$ body reference frame $\mathbb{B}_k$ as a material point on the body fixed to node $\mathbb{O}_k$ at the inboard hinge. For this choice, the assumed modes are cantilever modes, and node $\mathbb{O}_k$ exhibits zero deformation ($u_{nd}(\mathbb{O}_k) = \mathbf{0}$). Free–free modes are also used for representing body deformation, and are often preferred for control analysis and design. For these modes, the reference frame $\mathbb{B}_k$ is not fixed to any node, but is rather assumed to be fixed to the undeformed body, and, as a result, all nodes exhibit non-zero deformation. The dynamics models and algorithms developed here handle both types of modes.

Assume that a set of $n_{md}(k)$ assumed modes has been chosen for the $k^{th}$ body. Let $\Pi_r(\mathbb{O}_k^j) \in \mathcal{R}^6$ denote the **modal spatial displacement vector** at the $\mathbb{O}_k^j$ node for the $r^{th}$ mode, and $\eta(k) \in \mathcal{R}^{n_{md}(k)}$ denote the vector of modal deformation coordinates for the $k^{th}$ body. The $\Pi_r(\mathbb{O}_k^j) \in \mathcal{R}^6$ modal spatial displacement vectors are assumed to be constant and independent of the deformation of the body. The spatial deformation of node $\mathbb{O}_k^j$ is represented using modal coordinates as follows:

$$u_{nd}(\mathbb{O}_k^j) = \sum_{r=1}^{n_{md}(k)} \Pi_r(\mathbb{O}_k^j)\eta_r(k) \tag{19}$$

$\eta_r(k)$ denotes the $r^{th}$ element of $\eta(k)$. Eq. 19 defines a linear relationship between the modal coordinates and the deformation spatial displacements at the nodes. For cantilever modes,

$$\Pi_r(\mathbb{O}_k) = \mathbf{0} \quad \text{for} \quad r = 1 \cdots n_{md}(k) \tag{20}$$

Eq. 19 can be re-expressed as

$$u_{nd}(\mathbb{O}_k^j) = \Pi(\mathbb{O}_k^j)\eta(k) \tag{21}$$

where the **modal spatial displacement influence vector** $\Pi(\mathbb{O}_k^j)$ for the $\mathbb{O}_k^j$ node is defined as:

$$\Pi(\mathbb{O}_k^j) \triangleq \left[\Pi_1(\mathbb{O}_k^j), \cdots, \Pi_{n_{md}(k)}(\mathbb{O}_k^j)\right]_{r=1}^{n_{md}(k)} \in \mathcal{R}^{6 \times n_{md}(k)}$$

The $\Pi(k)$ **modal matrix** for the $k^{th}$ body is defined as follows:

$$\Pi(k) \triangleq \text{col}\left\{\Pi(\mathbb{O}_k^j)\right\}_{j=1}^{n_{nd}(k)} \in \mathcal{R}^{6n_{nd}(k) \times n_{md}(k)}$$

and relates the modal coordinates to the deformation field for the $k^{th}$ body as follows:

$$u_{nd}(k) \overset{21}{=} \Pi(k)\eta(k) \tag{22}$$

The $r^{th}$ column of $\Pi(k)$ is denoted $\Pi_r(k) \in \mathcal{R}^{6n_{nd}(k)}$, and is the **mode shape** for the $r^{th}$ assumed mode for the $k^{th}$ body.

The modal generalized coordinates parameterize the deformation of a body, while the hinge generalized coordinates parameterize the large angle articulation motion of the body. Taken together, they completely characterize

the motion of a flexible body. Therefore, define the $\vartheta(k)$ generalized coordinates and $\dot{\vartheta}(k)$ generalized velocities vectors for the $k^{th}$ body as

$$\vartheta(k) \triangleq \begin{bmatrix} \eta(k) \\ \theta(k) \end{bmatrix} \in \mathcal{R}^{\overline{N}(k)} \quad \text{and} \quad \dot{\vartheta}(k) \triangleq \begin{bmatrix} \dot{\eta}(k) \\ \dot{\theta}(k) \end{bmatrix} \in \mathcal{R}^{\overline{N}(k)} \tag{23}$$

where $\overline{N}(k) \triangleq n_{md}(k) + r_v(k)$ represents the overall number of velocity degrees of freedom associated with the $k^{th}$ body. We next summarize the equations of motion for such a system.

### 3.0.1 Velocity level

The velocity recursion equation is

$$\mathcal{V}_{fl}(k) = \Phi^*_{fl}(k+1,k)\mathcal{V}_{fl}(k+1) + H^*_{fl}(k)\,\dot{\vartheta}(k) \tag{24}$$

and the partitioned form of $H_{fl}(k)$ is

$$H_{fl}(k) = \begin{bmatrix} H_{Mfl}(k) \\ H_{Rfl}(k) \end{bmatrix} \tag{25}$$

Introducing a dummy variable $k'$, we can rewrite Eq. 24 as

$$\begin{aligned} \mathcal{V}_{fl}(k') = \mathcal{V}(\mathbb{O}_k) \quad &= \Phi^*_{fl}(k+1,k')\mathcal{V}_{fl}(k+1) + H^*(k)\,\dot{\theta}(k) \\ \mathcal{V}_{fl}(k) = \begin{bmatrix} \dot{\eta}(k) \\ \mathcal{V}(k) \end{bmatrix} \quad &= \Phi^*_{fl}(k',k)\mathcal{V}_{fl}(k') + H^*_{Mfl}(k)\,\dot{\eta}(k) \end{aligned} \tag{26}$$

where

$$\Phi_{fl}(k+1,k') \triangleq \mathcal{A}_{fl}(k+1)\phi(\mathbb{O}^+_k,\mathbb{O}_k) \quad \text{and} \quad \Phi_{fl}(k',k) \triangleq [0, \quad \phi(\mathbb{O}_k,k)]$$

with

$$\mathcal{A}_{fl}(k+1) = \begin{bmatrix} \Pi^*(\mathbb{O}^+_k) \\ \phi(k+1,\mathbb{O}^+_k) \end{bmatrix} \quad \text{and} \quad H_{Mfl}(k) \triangleq [I, \quad -\Pi^*_{\mathbb{B}}(\mathbb{O}_k)] \quad \text{where} \quad \Pi_{\mathbb{B}}(\mathbb{O}_k) = \phi^*(\mathbb{O}_k,k)\Pi(\mathbb{O}_k)$$

Conceptually, each flexible body is replaced by two new bodies. The first body is assigned the kinematical and mass/inertia properties of the flexible body and also its deformation generalized coordinates. The second virtual body is mass-less, and has zero extent. It is assigned the hinge generalized coordinates. The serial-chain now contains twice the number of bodies as the original one, with half the new bodies being fictitious ones. The tree digraph associated with the new system contains twice as many nodes. The new $H^*_{fl}$ operator now has the same number of columns, but twice the number of rows, as the original $H^*_{fl}$ operator. The new $\Phi_{fl}$ operator has twice as many rows and columns as the original one.

Note that Eq. 26 can be written as

$$\begin{aligned} \mathcal{V}(\mathbb{O}^+_k) &= \mathcal{A}^*_{fl}(k+1)\mathcal{V}_{fl}(k+1) = \phi^*(k+1,\mathbb{O}^+_k)\mathcal{V}(k+1) + \Pi(\mathbb{O}^+_k)\,\dot{\eta}(k+1) \\ \mathcal{V}_{fl}(k') &= \mathcal{V}(\mathbb{O}_k) = \phi^*(\mathbb{O}^+_k,\mathbb{O}_k)\mathcal{V}(\mathbb{O}^+_k) + H^*(k)\,\dot{\theta}(k) \\ \mathcal{V}(k) &= \phi^*(\mathbb{O}_k,k)\,[\mathcal{V}(\mathbb{O}_k) - \Pi(\mathbb{O}_k)\,\dot{\eta}(k)] \\ \mathcal{V}_{fl}(k) &= \begin{bmatrix} \dot{\eta}(k) \\ \mathcal{V}(k) \end{bmatrix} \end{aligned} \tag{27}$$

### 3.0.2 Acceleration level

At the acceleration level we have

$$\begin{aligned}
\alpha_{fl}(k') &= \alpha(\mathbb{O}_k) = \Phi^*_{fl}(k+1,k')\alpha_{fl}(k+1) + H^*(k)\ddot{\theta}(k) + a_{fl}(k+1,k') \\
\alpha_{fl}(k) &= \begin{bmatrix} \dot{\eta}(k) \\ \alpha(k) \end{bmatrix} = \Phi^*_{fl}(k',k)\alpha_{fl}(k') + H^*_{Mfl}(k)\ddot{\eta}(k) + a_{fl}(k',k)
\end{aligned} \tag{28}$$

Based on Eq. 27, we can re-express Eq. 28 as

$$\begin{aligned}
\alpha(\mathbb{O}^+_k) &= \phi^*(k+1,\mathbb{O}^+_k)\alpha(k+1) + \Pi(\mathbb{O}^+_k)\ddot{\eta}(k+1) + a(k+1,\mathbb{O}^+_k) \\
\alpha_{fl}(k') &= \alpha(\mathbb{O}_k) = \phi^*(\mathbb{O}^+_k,\mathbb{O}_k)\alpha(\mathbb{O}^+_k) + H^*(k)\ddot{\theta}(k) + a(\mathbb{O}^+_k,\mathbb{O}_k) \\
\alpha(k) &= \phi^*(\mathbb{O}_k,k)\ [\alpha(\mathbb{O}_k) - \Pi(\mathbb{O}_k)\ddot{\eta}(k)] + a(\mathbb{O}_k,k) \\
\alpha_{fl}(k) &= \begin{bmatrix} \ddot{\eta}(k) \\ \alpha(k) \end{bmatrix}
\end{aligned} \tag{29}$$

with

$$a_{fl}(k+1,k') = \phi^*(\mathbb{O}^+_k,\mathbb{O}_k)a(k+1,\mathbb{O}^+_k) + a(\mathbb{O}^+_k,\mathbb{O}_k) \quad \text{and} \quad a_{fl}(k',k) = \begin{bmatrix} 0 \\ a(\mathbb{O}_k,k) \end{bmatrix} \tag{30}$$

Note that $a_{fl}(k+1,k')$ contains an additional $H^*(k)\ddot{\theta}(k)$ term if the hinge is prescribed.

### 3.0.3 Inverse dynamics forces level

At the inter-body forces level

$$\begin{aligned}
f_{fl}(k) &= \begin{bmatrix} f^f_{fl}(k) \\ f^r_{fl}(k) \end{bmatrix} = \Phi_{fl}(k,k'-1)f_{fl}(k'-1) + M_{fl}(k)\alpha_{fl}(k) + b_{fl}(k) + \mathfrak{K}(k)\vartheta(k) \\
\mathcal{T}_m(k) &= 0 = H_{Mfl}(k)f_{fl}(k) = f^f_{fl}(k) - \Pi^*_{\mathbb{B}}(\mathbb{O}_k)f^r_{fl}(k) = f^f_{fl}(k') - \Pi^*(\mathbb{O}_k)f_{fl}(k') \\
f_{fl}(k') &= \Phi_{fl}(k',k)f_{fl}(k) = \phi(\mathbb{O}_k,k)\ f^r_{fl}(k) \\
\mathcal{T}_{pr}(k) &= H(k)f_{fl}(k')
\end{aligned} \tag{31}$$

### 3.0.4 ATBI matrices level

The mass matrix factorization and inversion expressions in Eq. 10 continue to apply here for flexible bodies though with appropriately redefined spatial operators. Similarly the recursive algorithms based in Eq. 11 continue to apply as well. For the ATBI matrices for the new $k^{th}$ "flexible" body we have (computed by

$$\begin{aligned}
\mathcal{P}_{fl}(k) &= \sum \Phi_{fl}(k,k'-1)\mathcal{P}^+_{fl}(k'-1)\Phi^*_{fl}(k,k'-1) + M_{fl}(k) \\
&= \begin{pmatrix} \mathcal{P}^{ff}_{fl} & \mathcal{P}^{fr}_{fl}(k) \\ \mathcal{P}^{rf}_{fl}(k) & \mathcal{P}^{rr}_{fl}(k) \end{pmatrix} = \begin{pmatrix} I + \sum \Pi^*(\mathbb{O}^+_k)\mathcal{P}^+_{fl}(k'-1)\Pi(\mathbb{O}^+_k) & \sum \Pi^*(\mathbb{O}^+_k)\mathcal{P}^+_{fl}(k'-1)\phi^*(k,\mathbb{O}^+_{k-1}) \\ \sum \phi(k,\mathbb{O}^+_{k-1})\mathcal{P}^+_{fl}(k'-1)\Pi(\mathbb{O}^+_k) & \sum \phi(k,\mathbb{O}^+_{k-1})\mathcal{P}^+_{fl}(k'-1)\phi^*(k,\mathbb{O}^+_{k-1}) + M(k) \end{pmatrix} \\
\mathcal{D}_{fl}(k) &= H_{Mfl}(k)\mathcal{P}_{fl}(k)H^*_{Mfl}(k) \\
&= \mathcal{P}^{ff}_{fl} - \mathcal{P}^{fr}_{fl}(k)\Pi_{\mathbb{B}}(\mathbb{O}_k) - \Pi^*_{\mathbb{B}}(\mathbb{O}_k)\mu_{fl}(k) \text{ where } \mu_{fl} \triangleq \mathcal{P}^{rf}_{fl}(k) - \mathcal{P}^{rr}_{fl}(k)\Pi_{\mathbb{B}}(\mathbb{O}_k) \\
\mathcal{G}_{fl}(k) &= \begin{bmatrix} \times \\ \mathcal{G}_{cr}(k) \end{bmatrix} \text{ where } \mathcal{G}_{cr}(k) = \mu_{fl}(k)\mathcal{D}^{-1}_{fl}(k) \\
\mathcal{P}^+_{fl}(k) &= \begin{pmatrix} \times & \times \\ \times & \mathcal{P}^+_{cr}(k) \end{pmatrix}, \text{ where } \mathcal{P}^+_{cr}(k) = \mathcal{P}^{rr}_{fl}(k) - \mathcal{G}_{cr}(k)\mu^*_{fl}(k)
\end{aligned} \tag{32}$$

For the ATBI matrices for the new $k'$ "rigid" body we have

$$\mathcal{P}_{fl}(k') = \phi(\mathbb{O}_k, k)\mathcal{P}^+_{cr}(k)\phi^*(\mathbb{O}_k, k)$$
$$\begin{cases} \textbf{if } \text{prescribed} \\ \quad \mathcal{P}^+_{fl}(k') = \mathcal{P}_{fl}(k') \\ \textbf{else} \\ \quad \mathcal{D}_{fl}(k') = H(k)\mathcal{P}_{fl}(k')H^*(k) \\ \quad \mathcal{G}_{fl}(k') = \mathcal{P}_{fl}(k')H^*(k)\mathcal{D}^{-1}_{fl}(k') \\ \quad \overline{\tau}_{fl}(k') = I - \mathcal{G}_{fl}(k')H(k) \\ \quad \mathcal{P}^+_{fl}(k') = \overline{\tau}_{fl}(k')\mathcal{P}_{fl}(k') \\ \textbf{end if} \end{cases} \tag{33}$$

### 3.0.5 ATBI filtering level

For the ATBI filter for the "flexible" body we have

$$\begin{aligned} \mathfrak{z}(k) &= \begin{bmatrix} \mathfrak{z}_m(k) \\ \mathfrak{z}_{cr}(k) \end{bmatrix} = \sum \mathcal{A}_{fl}(k)\phi(\mathbb{O}^+_{k-1}, \mathbb{O}_{k-1})\mathfrak{z}^+(k'-1) + \mathfrak{b}_{fl}(k) + \begin{bmatrix} \mathcal{P}^{fr}_{fl}(k) \\ \mathcal{P}^{rr}_{fl}(k) \end{bmatrix} \mathfrak{a}(\mathbb{O}_k, k) + \mathfrak{K}(k)\mathfrak{d}(k) \\ \epsilon(k) &= \mathcal{T}_m(k) - \mathfrak{z}_m(k) + \Pi^*_{\mathbb{B}}(\mathbb{O}_k)\mathfrak{z}_{cr}(k) \\ \nu(k) &= \mathcal{D}^{-1}_{fl}(k)\epsilon(k) \\ \mathfrak{z}^+(k) &= \begin{bmatrix} \times \\ \mathfrak{z}^+_{cr}(k) \end{bmatrix} \text{ where } \mathfrak{z}^+_{cr}(k) = \mathfrak{z}_{cr}(k) + \mathcal{G}_{cr}(k)\epsilon(k) \end{aligned} \tag{34}$$

For the ATBI filter for the "rigid" body we have

$$\mathfrak{z}(k') = \phi(\mathbb{O}_k, k)\mathfrak{z}^+_{cr}(k) + \mathcal{P}_{fl}(k')\mathfrak{a}_{fl}(k+1, k')$$
$$\begin{cases} \textbf{if } \text{prescribed} \\ \quad \mathfrak{z}^+(k') = \mathfrak{z}(k') \\ \textbf{else} \\ \quad \epsilon(k') = \mathcal{T}_{pr}(k) - H(k)\mathfrak{z}(k') \\ \quad \nu(k') = \mathcal{D}^{-1}_{fl}(k')\epsilon(k') \\ \quad \mathfrak{z}^+(k') = \mathfrak{z}(k') + \mathcal{G}_{fl}(k')\epsilon(k') \\ \textbf{end if} \end{cases} \tag{35}$$

### 3.0.6 ATBI smoother level

For the ATBI smoother for the "rigid" body we have (computed by `DartsHinge::_updateATBISmootherCache()`)

$$\alpha^+_{\mathrm{fl}}(k') = \phi^*(\mathbb{O}^+_k, \mathbb{O}_k)\mathcal{A}^*_{\mathrm{fl}}(k+1)\alpha_{\mathrm{fl}}(k+1) \tag{36}$$

$$= \phi^*(\mathbb{O}^+_k, \mathbb{O}_k)\left[\phi^*(k+1, \mathbb{O}^+_k)\alpha_{\mathrm{cr}}(k) + 1 + \Pi(\mathbb{O}^+_k)\ddot{\eta}(k+1)\right] \tag{37}$$

$$\begin{cases} \textbf{if prescribed} \\ \quad \mathfrak{f}_{\mathrm{fl}}(k') = \mathcal{P}^+_{\mathrm{fl}}(k')\alpha^+_{\mathrm{fl}}(k') + \mathfrak{z}^+(k') = \mathcal{P}_{\mathrm{fl}}(k')\alpha^+_{\mathrm{fl}}(k') + \mathfrak{z}(k') \\ \quad \mathcal{T}_{\mathrm{pr}}(k) = H(k)\mathfrak{f}_{\mathrm{fl}}(k') \\ \textbf{else} \\ \quad \ddot{\theta}(k) = \nu(k') - \mathcal{G}^*_{\mathrm{fl}}(k')\alpha^+_{\mathrm{fl}}(k') \\ \textbf{end if} \end{cases} \tag{38}$$

$$\alpha_{\mathrm{fl}}(k') = \alpha^+_{\mathrm{fl}}(k') + H^*(k)\ddot{\theta}(k) + \mathfrak{a}_{\mathrm{fl}}(k+1, k') \tag{39}$$

For the ATBI smoother for the "flexible" body we have

$$\begin{aligned} \alpha^+_{\mathrm{fl}}(k) &= \begin{bmatrix} 0 \\ \alpha^+_{\mathrm{cr}}(k) \end{bmatrix} \quad \text{where} \quad \alpha^+_{\mathrm{cr}}(k) = \phi^*(\mathbb{O}_k, k)\alpha_{\mathrm{fl}}(k') \\ \ddot{\eta}(k) &= \nu(k) - \mathcal{G}^*_{\mathrm{cr}}(k)\alpha^+_{\mathrm{cr}}(k) \\ \alpha_{\mathrm{fl}}(k) &= \begin{bmatrix} \ddot{\eta}(k) \\ \alpha_{\mathrm{cr}}(k) \end{bmatrix} \text{ where } \alpha_{\mathrm{cr}}(k) = \alpha^+_{\mathrm{cr}}(k) - \Pi_{\mathbb{B}}(\mathbb{O}_k)\ddot{\eta}(k) + \mathfrak{a}(\mathbb{O}_k, k) \end{aligned} \tag{40}$$

### 3.0.7 ATBI inter-body forces

For the inter-body forces, we have

$$\begin{aligned} \mathfrak{f}_{\mathrm{fl}}(k) &= \begin{bmatrix} \mathfrak{f}^{\mathrm{f}}_{\mathrm{fl}}(k) \\ \mathfrak{f}^{\mathrm{r}}_{\mathrm{fl}}(k) \end{bmatrix} = \mathcal{P}_{\mathrm{fl}}(k)(\alpha_{\mathrm{fl}}(k) - \mathfrak{a}_{\mathrm{fl}}(k', k)) + \mathfrak{z}(k) = \mathcal{P}^+_{\mathrm{fl}}(k)\alpha^+_{\mathrm{fl}}(k) + \mathfrak{z}^+(k) \\ \mathfrak{f}_{\mathrm{fl}}(k') &= \mathcal{P}_{\mathrm{fl}}(k')(\alpha_{\mathrm{fl}}(k') - \mathfrak{a}_{\mathrm{fl}}(k+1, k')) + \mathfrak{z}(k') = \mathcal{P}^+_{\mathrm{fl}}(k')\alpha^+_{\mathrm{fl}}(k') + \mathfrak{z}(k') \end{aligned} \tag{41}$$

Note that the second equation corresponds to the physically observable inter-body force at the hinge that we are interested in. The structure of this expression identical to the one for the rigid body case in Eq. 18 seen earlier for rigid-body systems. Once again, our earlier observations that the quantities needed to evaluate the inter-body force are by products of the articulated body inertia algorithm for solving the equations of motion.

# 4 Closed-Chain Multibody Systems

So far we have studied tree-topology multibody systems. We now turn to closed-chain topology systems, i.e. a tree topology sub-system subject to a set of closure constraints. Examples of such local loops include constraints associated with geared motors, 4-bar linkages/wishbone suspensions, differentials etc. The presence of the local loops implies that the system is no longer a tree-topology system. We focus on two approaches for working with such systems. The first, the *tree-augmented (TA)* approach decomposes the system into a maximal spanning tree based tree topology system, together with a minimal set of closure constraints. The *constraint-embedding (CE)* approach uses graph transformations to eliminate the explicit constraints and convert the closed-topology into a tree-topology system.

In this section we study the TA approach for closed-chain systems, and the problem of computing inter-body forces for them. The CE approach is studied in Section 5. The tree formulation is section 2 directly applies to the tree-topology system in the TA approach.

Let $n_c$ denote the dimensionality of the closure constraints on the system, Then there exists a $G(\theta,t) \in \mathcal{R}^{n_c \times \mathcal{N}}$ matrix and a $\mathfrak{U}(t) \in \mathcal{R}^{n_c}$ vector that defines the velocity domain constraint equation for the holonomic and non-holonomic closure constraints on the system as follows:

$$G(\theta,t)\ \dot{\theta} = \mathfrak{U}(t) \tag{42}$$

We assume that $G(\theta,t)$ is a *full-rank* matrix. Observe that Eq. 42 is linear in the $\dot{\theta}$ generalized velocity coordinates. These constraints effectively reduce the independent generalized velocities for the system from $\mathcal{N}$ to an $(\mathcal{N}-n_c)$ dimensional linear space,

The dynamics of closed-chain systems can be obtained by modifying the tree system dynamics to include the effect of the closure constraints via *Lagrange multipliers*, $\lambda \in \mathcal{R}^{n_c}$, as follows[1]

$$\begin{aligned} \mathcal{M}(\theta)\ddot{\theta} + \mathcal{C}(\theta,\ \dot{\theta}) - G^*(\theta,t)\lambda &= \mathcal{T} \\ G(\theta,t)\ \dot{\theta} &= \mathfrak{U}(t) \end{aligned} \tag{43}$$

The $-G^*(\theta,t)\lambda$ term in the first equation represents the internal generalized constraint forces from the closure constraints.

By differentiating the constraint equation, Eq. 43 can be rearranged into the following descriptor form:

$$\begin{pmatrix} \mathcal{M} & G^* \\ G & \mathbf{0} \end{pmatrix} \begin{bmatrix} \ddot{\theta} \\ -\lambda \end{bmatrix} = \begin{bmatrix} \mathcal{T}-\mathcal{C} \\ \acute{\mathfrak{U}} \end{bmatrix} \quad \text{where} \quad \acute{\mathfrak{U}} \triangleq \dot{\mathfrak{U}}(t) - \dot{G}\ \dot{\theta}\ \in \mathcal{R}^{n_c} \tag{44}$$

One approach to solving the closed-chain dynamics equations of motion is to assemble the matrix on the left and the vector on the right in Eq. 44 and solve the linear matrix equation for the $\ddot{\theta}$ generalized accelerations. We on the other hand pursue an alternative Schur complement-based solution approach for the TA models as described in the following lemma.

The closed-chain dynamics generalized accelerations in Eq. 44 can be expressed as

$$\ddot{\theta} = \ddot{\theta}_f + \ddot{\theta}_\delta \tag{45}$$

where, the free generalized accelerations, $\ddot{\theta}_f$, the correction generalized accelerations, $\ddot{\theta}_\delta$, and the Lagrange multipliers, $\lambda$, are given by

$$\ddot{\theta}_f \triangleq \mathcal{M}^{-1}(\mathcal{T}-\mathcal{C}) \tag{46a}$$

$$\lambda = -\left[G\mathcal{M}^{-1}G^*\right]^{-1}\gamma \quad \text{where} \quad \gamma \triangleq G\ddot{\theta}_f - \acute{\mathfrak{U}} \tag{46b}$$

$$\ddot{\theta}_\delta \triangleq \mathcal{M}^{-1}G^*\ \lambda \tag{46c}$$

The $\ddot{\theta}_f = \mathcal{M}^{-1}(\mathcal{T}-\mathcal{C})$ term represents the generalized accelerations solution for the dynamics of the tree system while ignoring the closure constraints and is therefore referred to as the *free* generalized accelerations. $\gamma$ represents the acceleration-level constraint violation resulting from just the free dynamics of the system. The $G\mathcal{M}^{-1}G^*$ matrix in Eq. 46b is the Schur complement of the matrix on the left hand side of Eq. 44. An intuitive interpretation of Eq. 46b is that the constraint error spatial accelerations from the free-dynamics solution, together with the Schur complement matrix allow the computation of the constraint forces necessary to nullify the errors. Once the constraint forces are available, Eq. 46c uses them to obtain the generalized accelerations to correct the free system dynamics solution. The solution to the closed-chain forward dynamics thus involves the following steps:

1. Solve Eq. 46a for the $\ddot{\theta}_f$ free generalized accelerations using the tree-topology articulated body inertia algorithm in section 2.2.

[1] For a matrix $A$, the $A^*$ notation denotes its matrix transpose.

2. Use $\ddot{\theta}_f$ and the $G\mathcal{M}^{-1}G^*$ Schur complement to solve for the $\lambda$ Lagrange multipliers via Eq. 46b.

3. Use $\lambda$ to solve Eq. 46c for the $\ddot{\theta}_\delta$ correction accelerations using the tree-topology articulated body inertia algorithm in section 2.2.

4. Compute the $\ddot{\theta}$ generalized accelerations using Eq. 45.

The TA closed-chain forward dynamics solution uses the tree ATB forward dynamics twice in Steps 1 and 3 respectively, and combines the results to obtain the complete solution. The expensive ATB matrix computations sweep computations from Step 1 can be reused in Step 3 and do not need to be repeated.

The previously developed method for computing the inter-body forces for tree systems from section 2.3 can be used in each of the articulated body inertia solutions, and combine their results to obtain the overall inter-body forces for the closed-chain system!

At the cut-joints, the inter-body forces are even simpler to obtain since the Lagrange multiplier values evaluated in Step 2 are precisely the inter-body forces for the cut-joint pair of bodies.

# 5 Constraint Embedding Dynamics

In this section we turn to using the constraint-embedding (CE) method for closed-chain topology systems. In this approach, the constraints are eliminated by using a graph transformation to convert the closed-graph system into a tree-topology system.

## 5.1 Constraint Sub-Groups

Let us now assume that within the multibody system, we have a a sub-graph of bodies with some closure constraints among them. One consequence of the constraints is that the effective degrees of freedom associated with the bodies in this sub-graph is less than the sum total of the hinge degrees of freedom. For simplicity, we assume that the system has only a single such local loop, and that it has only only a single child branch.

Our first step is to isolate this sub-graph in order to clearly define its internal kinematics/dynamics relationships, as well as its coupling to the rest of the system. For the purpose of exposition, we assume that the links have been numbered so that the indices for the links in the sub-graph range from from $i$ to $j$ with $i > j$. Assume that the $i^{th}$ link in the sub-graph is the child of the $(i+1)^{th}$ link and the $(j-1)^{th}$ link is the child of the $j^{th}$ link. We introduce a minimal number of cuts between the links internal to the sub-graph to convert it into a sub-tree. With these cuts, we can now think of the original system as being a tree-topology system with each sub-graph subject to additional internal constraints. Note that the location of the cuts is not unique in general. Putting aside the constraints for the moment, we are now in a position to bring to bear the full development to the tree-system in terms of defining the corresponding stacked vectors for generalized velocities $\dot{\theta}$, spatial velocities $\mathcal{V}$, the $\mathcal{E}_\phi$, $\phi$ and other spatial operators. While the operator identities and mass matrix factorization and inversions continue to apply for the tree-system, they do not describe the correct system dynamics since they do not take into account the internal constraints. Our next steps are to overcome this gap and embed the internal constraints into the tree system's dynamics so that they do indeed reflect the correct system dynamics.

## 5.2 Sub-graph Based Partitioning

Towards this, define the stacked spatial velocities vector $\mathcal{V}_\mathfrak{S} = \text{col}\left\{\mathcal{V}(j), \cdots \mathcal{V}(i)\right\}$, and the stacked generalized velocities vector $\dot{\theta}_\mathfrak{S} = \text{col}\left\{\dot{\theta}(j), \cdots \dot{\theta}(i)\right\}$ for the loop subgraph. The $\mathcal{V}_\mathfrak{S}$ and $\dot{\theta}_\mathfrak{S}$ vectors are sub-vectors of the

full $\mathcal{V}$ and $\dot{\theta}$ vectors corresponding to only the links in the sub-graph as follows:

$$\dot{\theta} = \begin{bmatrix} \dot{\theta}(1) \\ \vdots \\ \dot{\theta}(j-1) \\ \hline \dot{\theta}_{\mathfrak{S}} \\ \hline \dot{\theta}(i+1) \\ \vdots \\ \dot{\theta}(n) \end{bmatrix}, \qquad \mathcal{V} = \begin{bmatrix} \mathcal{V}(1) \\ \vdots \\ \mathcal{V}(j-1) \\ \hline \mathcal{V}_{\mathfrak{S}} \\ \hline \mathcal{V}(i+1) \\ \vdots \\ \mathcal{V}(n) \end{bmatrix} \tag{47}$$

The partitioning of $\dot{\theta}$ and $\mathcal{V}$ using the $\dot{\theta}_{\mathfrak{S}}$ and $\mathcal{V}_{\mathfrak{S}}$ sub-vectors actually carries over to all the stacked vectors and spatial operators for the tree system. Thus, the $\mathcal{E}_\phi$ operator can be re-expressed in partitioned form as:

$$\mathcal{E}_\phi = \left(\begin{array}{ccc|c|cc} 0 & 0 & 0 & \dots & 0 & \dots \; 0 & 0 \\ \phi(2,1) & 0 & \dots & \dots & 0 & \dots \; 0 & \\ 0 & \phi(3,2) & \dots & \dots & 0 & \dots \; 0 & 0 \\ \vdots & \vdots & \ddots & \dots & 0 & \dots \; \vdots & \vdots \\ \hline \vdots & \vdots & \phi(j,j-1) & & & \dots \; \vdots & \vdots \\ \vdots & \vdots & \dots & \multicolumn{2}{c|}{\mathcal{E}_{\phi_{\mathfrak{S}}}} & \dots \; \vdots & \vdots \\ \vdots & \vdots & 0 & & & \dots \; \vdots & \vdots \\ \hline \vdots & \vdots & \dots & \dots & \phi(i+1,i) & \ddots \; \vdots & \vdots \\ \vdots & \vdots & \ddots & \dots & 0 & \dots \; \ddots & \vdots \\ 0 & 0 & \dots & \dots & 0 & \dots \; \phi(\mathcal{N},\mathcal{N}-1) & 0 \end{array}\right) \tag{48}$$

Here $\mathcal{E}_{\phi_{\mathfrak{S}}}$ is the block element of $\mathcal{E}_\phi$ corresponding to just the bodies in the sub-graph.

### 5.3 Reformulated Velocity Relationships

Define the $\mathsf{E}_{\mathfrak{S}}$ and $\mathcal{B}_{\mathfrak{S}}$ connector operators as:

$$\begin{aligned} \mathsf{E}_{\mathfrak{S}} &\triangleq [0,0,\cdots \phi(i+1,i)] \\ \mathcal{B}_{\mathfrak{S}} &\triangleq = [\phi^*(j,j-1),0,\cdots 0]^* \end{aligned} \tag{49}$$

As can be observed from Eq. 5, $\mathsf{E}_{\mathfrak{S}}$ and $\mathcal{B}^*_{\mathfrak{S}}$ are also sub-blocks of $\mathcal{E}_\phi$ that denote the coupling of the sub-graph to links $i+1$ and $j-1$ respectively. Using these operators, We can now re-state the spatial velocity relationships between these links and the sub-graph as follows:

$$\begin{aligned} \mathcal{V}_{\mathfrak{S}} &= \mathcal{E}^*_{\phi_{\mathfrak{S}}} \mathcal{V}_{\mathfrak{S}} + \mathsf{E}^*_{\mathfrak{S}} \mathcal{V}(i+1) + \mathsf{H}^*_{\mathfrak{S}} \, \dot{\theta}_{\mathfrak{S}} \\ \mathcal{V}(j-1) &= \phi^*(j,j-1) \mathcal{B}^*_{\mathfrak{S}} \mathcal{V}_{\mathfrak{S}} + \mathsf{H}^*(j-1) \, \dot{\theta}(j-1) \end{aligned} \tag{50}$$

In essence, Eq. 50 is a block-partitioned restatement of the system level velocity relationships. The first equation defines how the parent $(i+1)^{th}$ body's spatial velocity couples into the sub-graph while the second one defines how the sub-graphs velocities couple into the outboard $(j-1)^{th}$ body. Continuing on, we now define sub-graph rigid body transformation operator $\phi_{\mathfrak{S}}$ analogous to Eq. 4 as

$$\phi_{\mathfrak{S}} \triangleq (I - \mathcal{E}_{\phi_{\mathfrak{S}}})^{-1} \tag{51}$$

Note that $\phi_{\mathcal{G}}$ is a sub-block of the full $\phi$ corresponding to the sub-graph's bodies. With this we can rewrite the first equation in Eq. 50 as

$$\mathcal{V}_{\mathcal{G}} \overset{50}{=} \phi^*_{\mathcal{G}} E^*_{\mathcal{G}} \mathcal{V}(i+1) + \phi^*_{\mathcal{G}} H^*_{\mathcal{G}} \dot{\theta}_{\mathcal{G}} \tag{52}$$

Comparing the structure of Eq. 50 and 51 with the velocity recursions in Eq. 2, we see that the parallels allow us to identify the stacked vectors and operators associated with the sub-graph as though they corresponded to a single **aggregate link**. From this perspective, the aggregate link's link transformation matrix is $\phi^*_{\mathcal{G}} E^*_{\mathcal{G}}$ and joint map matrix is $\phi^*_{\mathcal{G}} H^*_{\mathcal{G}}$. $\mathcal{V}_{\mathcal{G}}$ is the spatial velocity for the aggregate link, and its associated generalized velocity coordinates are $\dot{\theta}_{\mathcal{G}}$. In contrast with the other bodies that are all rigid, the aggregate link's configuration varies with the sub-graph's coordinates.

We can continue on in a similar vein to develop recursions that paralleling the remaining ones in Eq. 2 for the aggregate link. In effect, except for the constraints internal to the aggregate link, the structural equations for the system with the aggregate link have been transformed into a form resembling those for tree-topology systems. The next section addresses an issue that we have side-stepped so far - the constraints within the aggregate link

### 5.4 Embedding the sub-graph constraints

Having seen how to decompose and isolate sub-graphs of bodies as aggregate bodies we now turn to the subject of handling closure-constraints within these sub-graphs. We describe here the process of embedding these constraints directly into the dynamics model so that we can complete the transformation to the simpler tree topology model. Note that the loop constraint imposes internal consistency conditions on the elements of $\mathcal{V}_{\mathcal{G}}$, and these conditions are met by admissible $\dot{\theta}_{\mathcal{G}}$ generalized velocities that are consistent with the constraints.

Due to the internal constraints within the sub-graph, the elements of $\dot{\theta}_{\mathcal{G}}$ are not independent but are instead implicitly coupled by the constraint. At the coordinate level the constraint often takes the form of a nonlinear algebraic relationship. However, at the velocity level, the generalized velocities in the system are subject to a linear, though configuration dependent, constraint. Based on the rank of the constraint matrix, it is thus possible to partition the sub-graph's generalized velocities into sets of independent and dependent generalized velocities. The independent set is denoted $\dot{\theta}_{R\mathcal{G}}$ and is such that the dependent generalized velocities that satisfy the sub-graph constraint can be obtained from them. In other words, there exists a (configuration dependent) mapping $X_{\mathcal{G}}$ such that

$$\dot{\theta}_{\mathcal{G}} = X_{\mathcal{G}} \, \dot{\theta}_{R\mathcal{G}} \tag{53}$$

Thus we have

$$H^*_{\mathcal{G}} \, \dot{\theta}_{\mathcal{G}} = H^*_{R\mathcal{G}} \, \dot{\theta}_{R\mathcal{G}} \quad \text{where} \quad H^*_{R\mathcal{G}} \overset{\triangle}{=} H^*_{\mathcal{G}} X_{\mathcal{G}} \tag{54}$$

Using Eq. 54 in Eq. 52 we have the following new velocity transformation relationship for the sub-graph in terms of independent generalized velocities:

$$\mathcal{V}_{\mathcal{G}} \overset{51,52,54}{=} \phi^*_{\mathcal{G}} E^*_{\mathcal{G}} \mathcal{V}(i+1) + \phi^*_{\mathcal{G}} H^*_{R\mathcal{G}} \, \dot{\theta}_{R\mathcal{G}} = \phi^*_{\mathcal{G}} E^*_{\mathcal{G}} \mathcal{V}(i+1) + \underline{H}^*_{R\mathcal{G}} \, \dot{\theta}_{R\mathcal{G}} \tag{55}$$

where $\underline{H}^*_{R\mathcal{G}}$ is defined as

$$\underline{H}^*_{R\mathcal{G}} \overset{\triangle}{=} \phi^*_{\mathcal{G}} H^*_{R\mathcal{G}} \tag{56}$$

Eq. 55 depends only upon the independent generalized velocities for the sub-graph. It has in effect, eliminated the sub-graph's dependent generalized velocities by embedded the sub-graphs constraints directly into the relationship. In other words, the relationship is using the minimal set of generalized velocities which always satisfy the sub-graph's constraints. At least for the body velocities, we now have a complete tree-topology structure for the recursive relationships.

### 5.5 Acceleration-Level Relationships

Analogous expressions to Eq. 50 at the acceleration level take the form :

$$\begin{aligned} \alpha_{\mathcal{G}} &= \mathcal{E}^*_{\phi\,\mathcal{G}}\alpha_{\mathcal{G}} + E^*_{\mathcal{G}}\alpha(i+1) + H^*_{\mathcal{G}}\ddot{\theta}_{\mathcal{G}} + a_{\mathcal{G}} \\ \implies \alpha_{\mathcal{G}} &\overset{51}{=} \phi^*_{\mathcal{G}}E^*_{\mathcal{G}}\alpha(i+1) + \phi^*_{\mathcal{G}}H^*_{\mathcal{G}}\ddot{\theta}_{\mathcal{G}} + \phi^*_{\mathcal{G}}a_{\mathcal{G}} \end{aligned} \tag{57}$$

$\alpha_{\mathcal{G}}$ and $a_{\mathcal{G}}$ above correspond to the partitioned sub-vectors of $\alpha$ body spatial acceleration and $a$ Coriolis spatial acceleration stacked vectors for the tree-topology system. Differentiating Eq. 53 we have

$$\ddot{\theta}_{\mathcal{G}} = X_{\mathcal{G}}\ddot{\theta}_{R\mathcal{G}} + \dot{X}_{\mathcal{G}}\,\dot{\theta}_{R\mathcal{G}} \tag{58}$$

Using this in Eq. 57 leads to

$$\begin{aligned} \alpha_{\mathcal{G}} &\overset{57}{=} \phi^*_{\mathcal{G}}E^*_{\mathcal{G}}\alpha(i+1) + \underline{H}^*_{R\mathcal{G}}\ddot{\theta}_{R\mathcal{G}} + \phi^*_{\mathcal{G}}a_{\mathcal{G}} + \phi^*_{\mathcal{G}}H^*_{\mathcal{G}}\,\dot{X}_{\mathcal{G}}\,\dot{\theta}_{R\mathcal{G}} \\ &= \phi^*_{\mathcal{G}}E^*_{\mathcal{G}}\alpha(i+1) + \underline{H}^*_{R\mathcal{G}}\ddot{\theta}_{R\mathcal{G}} + \phi^*_{\mathcal{G}}a'_{\mathcal{G}} \end{aligned} \tag{59}$$

where

$$a'_{\mathcal{G}} \overset{\triangle}{=} a_{\mathcal{G}} + H^*_{\mathcal{G}}\,\dot{X}_{\mathcal{G}}\,\dot{\theta}_{R\mathcal{G}} \tag{60}$$

Note that the structure of Eq. 55 and Eq. 59 resemble those of the velocity and acceleration relations in Eq. 2 for tree-topology systems. We can continue on to transform the spatial force expressions as well. For this we will need the spatial inertia term, $M$, for the aggregate body. $M_{\mathcal{G}}$ is simply the partitioned sub-block of $M$ corresponding to the sub-graph. We skip the details since the development is similar and simply state the resulting expressions involving the aggregate body below:

$$\begin{aligned} f'_{\mathcal{G}} &= \mathcal{B}_{\mathcal{G}}\phi(j,j-1)f(j-1) + M_{\mathcal{G}}\alpha_{\mathcal{G}} + b_{\mathcal{G}} \\ f(i+1) &= E_{\mathcal{G}}\phi_{\mathcal{G}}f'_{\mathcal{G}} + M(i+1)\alpha(i+1) + b(i+1) \\ \mathcal{T}_{R\mathcal{G}} &= \underline{H}_{R\mathcal{G}}f'_{\mathcal{G}} \\ \mathcal{T}(i+1) &= H(i+1)f(i+1) \end{aligned} \tag{61}$$

Taken together, Eq. 60 and 61 define the parts of the equations of motion effected by the use of the aggregate link. *The equations of motion have the same structure as for tree-topology systems and moreover the constraints have been eliminated by embedding them directly into the dynamics formulation. The reduced generalized velocities vector using* $\dot{\theta}_{R\mathcal{G}}$ *has minimal dimension. The elimination of the constraints means that the equations of motion now form an ordinary differential equation whose solution is guaranteed to satisfy the constraints on the system in contrast with having to work with the significantly more complex differential-algebraic formulation we began with. However, the significant benefit from the new formulation is that this aggregate link based, tree-topology, minimal coordinate formulation meets all the requirements of the spatial operator analytical techniques including those leading to the mass matrix factorization and inversion expressions described in Eq. 10.* In effect, the new formulation is an instance of projected dynamics but one with the key advantage of preserving the tree-topology structure within the formulation. While we can now take advantage of the available spatial operator techniques to tackle a wide range of analysis and computational problems for systems with constraints, in the following sections we focus here on the important problem of extending the well-known $O(\mathcal{N})$ forward dynamics algorithms to the constrained dynamics case.

### 5.6 Forward Dynamics with Constraint Embedding

As noted above, using the aggregate link with constraint embedding, we have a tree-topology structure with just the reduced set of independent generalized velocities and forces. Since the new system and the mass matrix continue to have the familiar structure for tree-topology systems, we can repeat the steps leading to the Innovations

Factorization and inversion of the mass matrix in Eq. 10 for the new reduced mass matrix. One noteworthy change however is that some of the dimensionality of the sub-blocks associated with the aggregate link in the new operators are much larger than for normal rigid links. Indeed, the larger the sub-graphs, the larger is the size of these block-elements.

With this setup, all of the inverse and forward dynamics results in Eq. 11 to 7 continue to apply for the new operators defined by embedding the constraints. We now look in more detail at extending the $O(\mathcal{N})$ forward dynamics algorithm in Eq. 7 for the aggregate link formulation. Towards this, we simply need to replace the $\phi(.,.)$, $H^*(.)$ terms in Eq. 7 with the corresponding terms for the aggregate links to obtain a variant of the Riccati equation computations in the extended articulated body inertia recursive forward dynamics algorithm. We only describe below the steps for the aggregate link below since these are the only ones that are different from the steps in Eq. 7:

$$\begin{aligned}
\mathcal{P}^+(j-1) &= \overline{\tau}(j-1)\mathcal{P}(j-1) \\
\mathcal{P}_{\mathfrak{S}} &= \mathcal{B}_{\mathfrak{S}}\phi(j,j-1)\mathcal{P}^+(j-1)\phi^*(j,j-1)\mathcal{B}^*_{\mathfrak{S}} + \mathcal{M}_{\mathfrak{S}} \\
\mathcal{D}_{\mathfrak{S}} &= \underline{H}_{\mathcal{R}\mathfrak{S}}\mathcal{P}_{\mathfrak{S}}\underline{H}^*_{\mathcal{R}\mathfrak{S}} \\
\mathcal{G}_{\mathfrak{S}} &= \mathcal{P}_{\mathfrak{S}}\underline{H}^*_{\mathcal{R}\mathfrak{S}}\mathcal{D}^{-1}_{\mathfrak{S}} \\
\tau_{\mathfrak{S}} &= \mathcal{G}_{\mathfrak{S}}\underline{H}_{\mathcal{R}\mathfrak{S}} \\
\mathcal{P}^+_{\mathfrak{S}} &= \mathcal{P}_{\mathfrak{S}} - \tau_{\mathfrak{S}}\mathcal{P}_{\mathfrak{S}} \\
\mathcal{P}(i+1) &= E_{\mathfrak{S}}\phi_{\mathfrak{S}}\mathcal{P}^+_{\mathfrak{S}}\phi^*_{\mathfrak{S}}E^*_{\mathfrak{S}} + M(i+1)
\end{aligned} \tag{62}$$

Note that $\mathcal{D}_{\mathfrak{S}}$ can be re-expressed as

$$\mathcal{D}_{\mathfrak{S}} = X_{\mathfrak{S}}{}^*(H_{\mathfrak{S}}\phi_{\mathfrak{S}}\mathcal{P}_{\mathfrak{S}}\phi^*_{\mathfrak{S}}H^*_{\mathfrak{S}})X_{\mathfrak{S}} = X_{\mathfrak{S}}{}^*\mathcal{M}_{\mathfrak{S}}X_{\mathfrak{S}} \quad \text{where} \quad \mathcal{M}_{\mathfrak{S}} \triangleq H_{\mathfrak{S}}\phi_{\mathfrak{S}}\mathcal{P}_{\mathfrak{S}}\phi^*_{\mathfrak{S}}H^*_{\mathfrak{S}} \tag{63}$$

The inner term $\mathcal{M}_{\mathfrak{S}}$ has the structure of the mass matrix of the sub-graph's tree. The one major difference from the true sub-tree's mass matrix is that the central body spatial inertia operator is $\mathcal{P}_{\mathfrak{S}}$ instead of the normal $M_{\mathfrak{S}}$ term. However, the structural properties of a tree-topology continue to hold as do results such as the composite rigid body decomposition, operator-based mass matrix factorization and inversion etc. $\mathcal{D}_{\mathfrak{S}}$ is the reduced mass matrix for the aggregate link's sub-tree projected down down to its independent degrees of freedom. Link $(i+1)$ serves as the root of the aggregate link's sub-tree. Also, note that $\mathcal{P}_{\mathfrak{S}}$ includes the articulated body contribution from the sub-tree rooted at link $j-1$.

The accompanying vector recursions for the articulated body inertia algorithm in Eq. 7 take the following form for the aggregate link. First the step from body $(j-1)$ to the $i^{th}$ aggregate link:

$$\begin{aligned}
\mathfrak{z}^+(j-1) &= \mathfrak{z}(j-1) + \mathcal{G}(j-1)\epsilon(j-1) \\
\mathfrak{z}_{\mathfrak{S}} &= \mathcal{B}_{\mathfrak{S}}\phi(j,j-1)\mathfrak{z}^+(j-1) + \mathfrak{b}_{\mathfrak{S}} + \mathcal{P}_{\mathfrak{S}}\mathfrak{a}'_{\mathfrak{S}} \\
\epsilon_{\mathfrak{S}} &= \mathcal{T}_{\mathfrak{S}} - \underline{H}_{\mathcal{R}\mathfrak{S}}\mathfrak{z}_{\mathfrak{S}} \\
\nu_{\mathfrak{S}} &= \mathcal{D}^{-1}_{\mathfrak{S}}\epsilon_{\mathfrak{S}}
\end{aligned} \tag{64}$$

The recursion step from the $i^{th}$ body to body $(j+1)$ is as follows:

$$\begin{aligned}
\mathfrak{z}^+_{\mathfrak{S}} &= \mathfrak{z}_{\mathfrak{S}} + \mathcal{G}_{\mathfrak{S}}\epsilon_{\mathfrak{S}} \\
\mathfrak{z}(i+1) &= E_{\mathfrak{S}}\phi_{\mathfrak{S}}\mathfrak{z}^+_{\mathfrak{S}} + \mathfrak{b}(i+1) + \mathcal{P}(i+1)\mathfrak{a}(i+1) \\
\epsilon(i+1) &= \mathcal{T}(i+1) - H(i+1)\mathfrak{z}(i+1) \\
\nu(i+1) &= \mathcal{D}^{-1}(i+1)\epsilon(i+1)
\end{aligned} \tag{65}$$

The base-to-tip accelerations sweep steps also are also altered for the aggregate link. First the steps from body

$(i+1)$ to the $i^{th}$ aggregate body:

$$\begin{aligned}
\alpha^+_{\mathfrak{S}} &= \phi^*_{\mathfrak{S}} E^* \alpha(i+1) \\
\ddot{\theta}_{R\mathfrak{S}} &= \nu_{\mathfrak{S}} - \mathcal{G}_{\mathfrak{S}}{}^* \alpha^+_{\mathfrak{S}} \\
\alpha_{\mathfrak{S}} &= \alpha^+_{\mathfrak{S}} + \underline{H}^*_{R\mathfrak{S}} \ddot{\theta}_{R\mathfrak{S}} + a'_{\mathfrak{S}}
\end{aligned} \tag{66}$$

The step from the $i^{th}$ aggregate body to body $(j-1)$ is as follows:

$$\begin{aligned}
\alpha^+(i-1) &= \mathcal{B}^* \alpha_{\mathfrak{S}} \\
\ddot{\theta}(i-1) &= \nu(i-1) - \mathcal{G}^*(i-1)\alpha^+(i-1) \\
\alpha(i-1) &= \alpha^+(i-1) + H^*(i-1)\ddot{\theta}(i-1) + a(i-1)
\end{aligned} \tag{67}$$

The overall structure of the forward dynamics algorithm remains unchanged from the well-known tree-topology version with the only changes occurring at the steps involving the aggregate links. The most expensive part of these steps is the computation and inversion of the $\mathcal{D}_{\mathfrak{S}}$ square symmetric, positive definite matrix whose size is the number of degrees of freedom in the associated aggregate link. Thus the computational cost of the algorithm is no longer linear in the number of independent degrees of freedom for the aggregate links but is instead (in the worst case) quadratic in the total degrees of freedom in the aggregate links sub-tree and cubic in the number of independent degrees of freedom in the sub-tree. These additional costs however are modest when the loops are of moderate size.

### 5.7 Inter-body Forces

In the CE method, the system has a tree-topology after a graph transformation. Thus the tree-topology method described in section 2.3 can be directly used here. We however, take a closer look at the resulting expressions for a compound body since that part is different for a CE model. For a compound body, it follows from Eq. 18 that the inter-body force is given by

$$f'_{\mathfrak{S}} = \mathcal{P}_{\mathfrak{S}}(\alpha_{\mathfrak{S}} - a_{\mathfrak{S}}) + z_{\mathfrak{S}} = \mathcal{P}^+_{\mathfrak{S}} \alpha^+_{\mathfrak{S}} + z^+_{\mathfrak{S}} \tag{68}$$

However the physical inter-body forces for all the bodies in the aggregated subgraph are defined by the components of $f_{\mathfrak{S}}$ which is related to $f'_{\mathfrak{S}}$ via

$$f_{\mathfrak{S}} = \phi_{\mathfrak{S}} f'_{\mathfrak{S}} \tag{69}$$

The above expression $\phi_{\mathfrak{S}} f'_{\mathfrak{S}}$ can be computed recursively in a gather sweep over just the bodies in the compound body without having to compute a matrix/vector product of assembling $\phi_{\mathfrak{S}}$. Once again we see that the inter-body forces can be computed without excessive cost once the equations of motion have been solved.

## 6 Conclusions

This paper examines the topic of computing inter-body forces when using recursive minimal coordinate methods for solving the equations of motion. It is shown that the absence of Lagrange multipliers does not necessitate the use of inverse dynamics algorithms for computing the inter-body forces. Instead, a simple algebraic expression using the by-products of the articulated body inertia recursive dynamics algorithm yields the inter-body constraint force. While initially developed for tree-topology rigid body systems, similar expressions are shown to hold for the more complex case involving flexible body systems. For closed-loop systems, two methods are examined. The TA method is based on the well-known method of decomposing the system into a tree system with additional constraint, while the CE method uses graph transformations to eliminate the constraints. For both of these methods the inter-body force can be computed using variants of the method for tree-topology systems. In summary, we show that computing these inter-body forces is a low-cost operation even when using minimal coordinate recursive methods.

## Acknowledgments

The research described in this paper was performed at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration.[2]

## References

[1] A. Jain, *Robot and Multibody Dynamics: Analysis and Algorithms*. Springer, 2011.

[2] R. Featherstone, *Rigid Body Dynamics Algorithms*. Springer Verlag, 2008.

[3] A. Jain, "Unified formulation of dynamics for serial rigid multibody systems," *Journal of Guidance Control and Dynamics*, vol. 14, pp. 531–542, nov 1991.

[4] A. Jain and G. Rodriguez, "Recursive Flexible Multibody System Dynamics Using Spatial Operators," *Journal of Guidance Control and Dynamics*, vol. 15, no. 6, pp. 1453—-1466, 1992.

[5] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "Spatial operator algebra for multibody system dynamics," *Journal of the Astronautical Sciences*, vol. 40, no. 1, pp. 27–50, 1992.

[6] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "On-line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, jun 1980.

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

[8] G. Rodriguez, A. Jain, and K. Kreutz-Delgado, "A spatial operator algebra for manipulator modeling and control," *International Journal of Robotics Research*, vol. 10, no. 4, p. 371, 1991.

[9] G. Rodriguez and K. Kreutz-Delgado, "Spatial Operator Factorization and Inversion of the Manipulator Mass Matrix," *IEEE Transactions on Robotics and Automation*, vol. 8, pp. 65–76, feb 1992.

[10] G. Rodriguez, "Random Field Estimation Approach to Robot Dynamics," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, pp. 1081–1093, sep 1990.

[11] A. Jain and G. Rodriguez, "Diagonalized Lagrangian robot dynamics," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 4, pp. 571–584, 1995.

[12] A. Jain and G. Rodriguez, "Multibody Mass Matrix Sensitivity Analysis Using Spatial Operators," *Journal for Multiscale Computational Engineering*, vol. 1, no. 2-3, 2003.

[13] A. Jain and G. Rodriguez, "Linearization of manipulator dynamics using spatial operators," *Ieee Transactions On Systems Man And Cybernetics*, vol. 23, no. 1, pp. 239–248, 1993.

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